

WebSphere Commerce Portal V5.4 Solutions

Integrating WebSphere Commerce V5.4, Business Edition and WebSphere Portal V4.2

WebSphere Commerce portal architecture and design guidelines

Implement a commerce portal runtime and development environment

Working example for developing a commerce enabled portal B2B site



John Ganci
Sanjoy Banik
Normunds Saumanis
Kai Schwidder



International Technical Support Organization

**WebSphere Commerce Portal V5.4 Solutions
Integrating WebSphere Commerce V5.4, Business
Edition and WebSphere Portal V4.2**

June 2003

Note: Before using this information and the product it supports, read the information in "Notices" on page xi.

First Edition (June 2003)

This edition applies to WebSphere Commerce V5.4.0.5, Business Edition for Windows 2000, IBM Commerce Enhancement Pack - April 2003 Edition, and WebSphere Portal V4.2.1 Enable for Multiplatforms.

© Copyright International Business Machines Corporation 2003. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	xi
Trademarks	xii
Preface	xiii
The team that wrote this redbook	xiii
Become a published author	xv
Comments welcome	xvi
Part 1. Introduction to IBM WebSphere Commerce Portal	1
Chapter 1. Introduction to commerce enabled portals	3
1.1 Overview	4
1.1.1 Key concepts of portals	4
1.1.2 Benefits of a commerce enabled portal	8
1.1.3 Features of WebSphere Commerce Portal	9
1.1.4 WebSphere Commerce Portal component packaging	11
1.1.5 Prerequisite knowledge and information sources	12
1.2 Structure of this redbook	15
Chapter 2. WebSphere Commerce Portal architecture	17
2.1 WebSphere Commerce Portal architecture	18
2.1.1 Commerce enabled portal integration approaches	18
2.1.2 XML-based commerce enabled portal solution outlook	21
2.1.3 WebSphere Commerce Portal runtime environment	22
2.1.4 WebSphere Commerce Portal component architecture	23
2.1.5 WebSphere Commerce Portal HTTP request flow	26
2.1.6 WebSphere Commerce Portal architecture of key features	29
2.2 WebSphere Commerce and WebSphere Portal architecture overview ..	37
2.2.1 WebSphere Commerce architecture	37
2.2.2 WebSphere Portal architecture	46
2.3 Mobile client access	51
2.3.1 Where to find more information	55
2.4 Authentication and single sign-on	56
2.5 Access control	57
2.6 Session management	59
2.7 Presentation services and page rendering	61
Chapter 3. Patterns for e-business and commerce enabled portals	63
3.1 Patterns for e-business	64

3.1.1	Introduction to Patterns for e-business	64
3.1.2	Patterns for e-business layered asset model	66
3.1.3	Patterns for e-business Web site	67
3.2	Electronic Commerce composite pattern	67
3.2.1	Defining the Electronic Commerce composite pattern	68
3.2.2	Identified patterns	73
3.2.3	Summary	74
3.3	Selecting the Application pattern	75
3.3.1	Access Integration::Web Single Sign-On application pattern	75
3.3.2	Access Integration::Pervasive Device Access application pattern	77
3.3.3	Access Integration::Personalized Delivery application pattern	79
3.3.4	Self-Service::Directly Integrated Single Channel application pattern	80
3.3.5	Collaboration::Store and Retrieve application pattern	82
3.3.6	Collaboration::Directed Collaboration application pattern	84
3.3.7	Information Aggregation::Population Single-Step application pattern	86
3.3.8	Information Aggregation::Population Crawling and Discovery application pattern	88
3.3.9	Application Integration::Direct Connection application pattern	91
3.4	Review Runtime patterns	92
3.5	Review Product mappings	94
Part 2.	Working example scenario: commerce enabled portal	97
Chapter 4.	Business requirements analysis and solution design	99
4.1	Business scenario	100
4.1.1	Initial context	100
4.1.2	Business challenges	101
4.2	Requirements	102
4.2.1	Functional requirements	102
4.2.2	Nonfunctional requirements	105
4.3	Use case model	113
4.3.1	Use case overview	113
4.3.2	Front-end experience use cases detail	116
4.3.3	System use cases detail	122
4.3.4	Administration use cases detail	123
4.4	Architecture	125
4.4.1	Architecture overview	125
4.4.2	Architectural decisions	128
4.4.3	Component model	132
4.4.4	Operational model	134
Chapter 5.	Implement the runtime environment	165
5.1	Planning for implementation and deployment	166
5.1.1	Network environment	166

5.1.2	Hardware used within the ITSO test environment	167
5.1.3	Software used within the ITSO test environment	168
5.1.4	VMWare or Ghost	171
5.1.5	High-level implementation procedure	172
5.1.6	Downloading the IBM Commerce Enhancement Pack - April 2003 Edition	172
5.2	WebSphere Commerce node implementation	173
5.2.1	Windows 2000 Server installation	174
5.2.2	DB2 Server installation	174
5.2.3	IBM HTTP Server installation	176
5.2.4	WebSphere Application Server installation	177
5.2.5	WebSphere Commerce installation	181
5.2.6	WebSphere Commerce Payments installation	185
5.2.7	Commerce Enhancement Pack installation	188
5.2.8	WebSphere Commerce instance creation	190
5.2.9	Commerce Enhancement Pack post-install configuration	191
5.2.10	WebSphere Commerce administration tools verification	193
5.2.11	WebSphere Commerce Payments configuration	194
5.2.12	Database backup	196
5.2.13	Enable WebSphere Commerce portal adapter	196
5.2.14	Disable the WebSphere Commerce cache	197
5.2.15	Commerce Enabled Portal - APAR JR18068	198
5.3	Directory Server (LDAP) node implementation	199
5.3.1	Windows 2000 Server installation	199
5.3.2	DB2 Server installation	200
5.3.3	IBM HTTP Server installation	200
5.3.4	IBM Directory Server installation	200
5.3.5	IBM Directory Server configuration	202
5.3.6	IBM Directory Server verification	204
5.3.7	Import data for WebSphere Portal and WebSphere Commerce	205
5.3.8	Configure WebSphere Commerce for LDAP	210
5.3.9	Enable SSL between WebSphere Commerce and LDAP	214
5.4	WebSphere Portal node implementation	214
5.4.1	Windows 2000 Server installation	215
5.4.2	DB2 Server installation	215
5.4.3	IBM HTTP Server installation	216
5.4.4	WebSphere Application Server installation	216
5.4.5	WebSphere Portal installation	217
5.4.6	Commerce Enhancement Pack configuration	225
5.5	Enabling single sign-on between WebSphere Portal and WebSphere Commerce	228
5.5.1	Overview on single sign-on (SSO)	228
5.5.2	WebSphere Portal configuration for SSO	231

5.5.3	WebSphere Commerce configuration for SSO	236
5.5.4	Verify single sign-on (SSO) configuration	243
5.6	Deploying the ITSO B2B CEP store	245
5.7	DB2 Server node implementation	245
5.7.1	Windows 2000 Server installation	245
5.7.2	DB2 Server installation	246
5.7.3	Verify the DB2 instance TCP/IP connection port	247
5.7.4	Migrate WebSphere Commerce databases to remote DB2 Server	248
5.7.5	Migrate WebSphere Portal database to remote DB2 Server	250
5.8	Remote Web server node implementation	250
Chapter 6.	Implement the development environment	253
6.1	Development environment configuration options	254
6.1.1	Motivation for an end-to-end development environment	254
6.1.2	Solution overview for an end-to-end development environment	255
6.1.3	Development environment configurations	258
6.1.4	Hardware and software used in the development environment	260
6.2	Prerequisite software installation	261
6.2.1	DB2 Server installation	262
6.2.2	WebSphere Studio Application Developer installation	262
6.2.3	Jakarta Ant overview	265
6.3	WebSphere Commerce and WebSphere Studio Application Developer configuration	265
6.3.1	Prerequisite software	265
6.3.2	Commerce Enhancement Pack Toolkit installation	266
6.3.3	Create the WebSphere Commerce development instance	266
6.3.4	Publishing stores using Store Services (seed environment)	267
6.4	WebSphere Portal and WebSphere Studio Application Developer configuration	267
6.4.1	Prerequisite software	267
6.4.2	WebSphere Portal Toolkit V4.2.5 and WebSphere Portal V4.2.1 test environment installation	268
6.4.3	WebSphere Portal test environment server configuration	270
6.5	Integrating WebSphere Portal and WebSphere Commerce within WebSphere Studio Application Developer	274
6.5.1	Extract SSL certificate from WebSphere Commerce development environment	274
6.5.2	Enable SSL for WebSphere Portal and import the SSL certificate from WebSphere Commerce to WebSphere Portal	274
6.5.3	Commerce Enhancement Pack personalization installation on WebSphere Studio Application Developer 5 WebSphere Portal Test Environment (optional)	275
6.5.4	WebSphere Portal content publishing wizard installation (optional)	284

6.5.5 Deploy the ITSO CEP B2B sample	284
Chapter 7. Create a commerce enabled portal store	285
7.1 ITSO sample code	286
7.1.1 Procedure to download ITSO sample code	286
7.1.2 Description of ITSO sample code	286
7.2 Create the ITSO B2B CEP store template	287
7.2.1 Create a new project	287
7.2.2 Update core data	288
7.2.3 Update configuration data	290
7.3 Customizing the ITSO B2B CEP store	291
7.3.1 JavaScript naming convention	291
7.3.2 Form naming conventions	292
7.3.3 Stripping HTML tags	294
7.3.4 Enable Portal URI	294
7.3.5 Developing commerce enabled portlets	296
7.4 Package SAR and publish Web assets	296
7.4.1 Create a SAR using ant_create_sar.xml	297
7.4.2 Publish Web assets to WebSphere Commerce using ant_publish_WCS.xml	298
Chapter 8. Create portlets with Dynamic Context Groups	301
8.1 Dynamic Context Groups	302
8.1.1 Defining Dynamic Context Groups	302
8.1.2 Modifications to commerce portal JSPs	305
8.1.3 WebSphere Studio Application Developer tooling for creating Dynamic Context Groups	306
Chapter 9. Deploy the commerce enabled portal	309
9.1 Deploy the ITSO B2B CEP store to the runtime environment	310
9.1.1 ITSO sample code	311
9.1.2 Database backup	311
9.1.3 Server startup	312
9.1.4 Create a new organization for the ITSO B2B CEP store	312
9.1.5 Create a new user and add to CEP organization	314
9.1.6 Assign roles to an organization	316
9.1.7 Assign roles to a user of an organization	317
9.1.8 Prepare the ITSO B2B CEP store SAR for publishing	319
9.1.9 Create a store archive from the ITSO B2B CEP store template	320
9.1.10 Publish the store from Store Services	321
9.1.11 Copy commerce portal tools JSPs and update properties	321
9.1.12 Verify the ITSO B2B CEP store	323
9.1.13 Deploy the commerce enabled portlets	323
9.1.14 Set commerce portlet permissions	329

9.1.15	Verify the ITSO B2B CEP store portal functionality	331
9.2	Deploy the ITSO B2B CEP store to the development test environment	332
9.2.1	ITSO sample code	333
9.2.2	Database backup	333
9.2.3	Create CEP organization	333
9.2.4	Prepare the ITSO B2B CEP store template for publishing	334
9.2.5	Create the CEP store from the ITSO B2B CEP store template	335
9.2.6	Publish the CEP store	336
9.2.7	Post-publish CEP store configuration	337
9.2.8	Deploy the commerce enabled portlets	338
9.2.9	Verify commerce enabled portal	345
Chapter 10.	Creating a commerce enabled portal for mobile clients	347
10.1	Introduction	348
10.1.1	WAP	349
10.2	Deploying the WAP portal adapter	350
10.3	OpenWave.SDK 6.2 for WAP toolkit and simulator	352
10.4	Sample WAP WML commerce portlet JSPs	353
Part 3.	Appendixes	355
Appendix A.	WebSphere Portal and WebSphere Commerce sample LDIF files	357
	WebSphere Portal LDIF sample	358
	WebSphere Commerce LDIF sample	359
Appendix B.	Logon for commerce portlets without single sign-on	361
	Architectural decisions	362
	Modifications to Commerce Enhancement Pack code	364
	Known limitations	366
	Updating the WebSphere Studio Application Developer plug-in with the modified classes	367
	Updating B2B Direct sample with the modified classes	367
	Configuring WebSphere Commerce logon URL and password	368
Appendix C.	Tips and troubleshooting for commerce enabled portals	369
	Resolving issues during installation or testing	370
	Cannot start portal using Application Developer	370
	Portal cannot connect to the database during installation	371
	404 - File not found exception when accessing portal	372
	The system could not log into your account when running the portlet on the server	372
	Error occurs when publishing the portlet application	372
	Portlet page not found	372

WebSphere Portal installation and configuration	372
Troubleshooting a store publishing failure	373
WebSphere Commerce log files	373
Unlocking a store archive	373
Republish a contract or account	373
Publishing is successful but store does not display or does not function properly	374
Transaction log for the database is full	374
java.lang.OutOfMemory error	375
Back up a DB2 database	375
Restore a DB2 database	376
Reset a disable account	376
Appendix D. Additional material	377
Locating the Web material	377
Using the Web material	377
System requirements for downloading the Web material	378
How to use the Web material	378
Related publications	379
IBM Redbooks	379
Other resources	379
Referenced Web sites	380
How to get IBM Redbooks	381
IBM Redbooks collections	381
Index	383

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.


This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server™	DB2®	Quickplace®
@server™	Everyplace®	Redbooks™
Redbooks (logo)™ 	IBM®	Sametime®
developerWorks®	Lotus®	SecureWay®
ibm.com®	MQSeries®	Tivoli®
iSeries™	NetView®	VisualAge®
xSeries®	NetVista™	WebSphere®
AIX®	Notes®	
Domino™	OS/390®	
DB2 Universal Database™	OS/400®	

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Preface

Portals provide the user with a single point of access to a wide variety of content, data and services throughout an enterprise. The content displayed in portlets on the portal page can be personalized based on user preferences, site design and marketing campaigns.

IBM WebSphere Commerce Portal provides an integrated solution for aggregating and personalizing commerce content and data by leveraging the technologies of WebSphere Commerce V5.4 and WebSphere Portal V4.2. When using the WebSphere Commerce Portal, users have a single point of access to personalized content in portlets from Web browser clients and mobile devices.

This IBM Redbook provides an overview of the WebSphere Commerce Portal benefits, features and architecture. The focus of the redbook is an end-to-end working example scenario for a commerce enabled portal B2B store. The working example includes the following topics:

- ▶ Business requirements analysis and solution design
- ▶ Multi-tiered runtime implementation procedures
- ▶ Development environment with source-level debug of commerce portlets and JSPs
- ▶ Creating a commerce enabled portal store using WebSphere Studio Application Developer
- ▶ Creating Dynamic Context Groups
- ▶ Mobile client access

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.



The IBM Redbook team (Kai Schwidder, John Ganci, Sanjoy Banik, Normunds Saumanis)

John Ganci is a Senior Software Engineer, WebSphere® Specialist at the IBM ITSO, Raleigh Center. He writes extensively and teaches classes on WebSphere and related topics. John has 14 years of experience in product and application design, development, system testing, and consulting. His areas of expertise include e-commerce, pervasive computing, and Java™ programming. Before joining the ITSO, he developed e-commerce sites for IBM.

Sanjoy Banik is a Principal Consultant with Gen-i Limited in New Zealand. He has five years of experience in electrical and electronics engineering, including the electro-medical sector, and 11 years of experience in I/T fields, including teaching computer science. He holds a Master's degree in Computer Science from University of Auckland and a Bachelor's degree in Electrical and Electronics Engineering from Bangladesh Institute of Technology. His areas of expertise include Solutions Architecture and Development, Training, Mentoring, Technical Team Leadership, e-business and portal solutions with Java and Microsoft® architecture. He has written three Redbooks on e-commerce and portal.

Normunds Saumanis is an I/T Architect in IBM Global Services, Latvia. He has over 10 years experience in systems support, systems integration, application development and I/T consulting. He holds a degree in Computer Science from Michigan State University, USA. His areas of expertise include AIX/UNIX®

systems support, IT infrastructure design and operations, systems integration, Java, pervasive and Web applications, IBM® WebSphere.

Kai Schwidder is an IT Architect in IBM Software Group, Switzerland. He has 13 years of experience in the field of consulting, application development, and systems integration for e-business and e-commerce solutions. He holds a degree in Computer Science from the Technical University in Berlin, Germany. His areas of expertise include systems integration, application architecture and development, business to technology consulting, technical team leadership, WebSphere Commerce, WebSphere Portal and WebSphere MQ.

Special thanks to **Joanna Ng** of IBM Canada, Architect and Manager of WebSphere Commerce Portal Development.

Thanks also to the following people for their contributions to this project:

Vincent Watt, IBM Canada

John Wenzel, IBM US

Terry Hudson, IBM UK

Norm Smith, IBM US

Gail Christensen, IBM US Raleigh ITSO Center

Linda Robinson, IBM US Raleigh ITSO Center

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

ibm.com/redbooks

- Send your comments in an Internet note to:

redbook@us.ibm.com

- Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 662
P.O. Box 12195
Research Triangle Park, NC 27709-2195



Introduction to IBM WebSphere Commerce Portal



Introduction to commerce enabled portals

This IBM Redbook describes the features and architecture of commerce enabled portals delivered in the IBM Commerce Enhancement Pack - April 2003 Edition for use with WebSphere Commerce V5.4.0.5 and WebSphere Portal V4.2.1. The redbook includes a working example scenario and sample code that provides an end-to-end solution for designing, developing, and deploying a commerce enabled portal Web site.

In this chapter, we introduce the concepts and key components of a commerce enabled portal. After reading this chapter, you should have a clear understanding of the following topics relevant to commerce enabled portals:

- ▶ Key concepts of portals
- ▶ Benefits of a commerce enabled portal
- ▶ Features of WebSphere Commerce Portal
- ▶ WebSphere Commerce Portal component packaging
- ▶ Prerequisite knowledge and information sources
- ▶ Structure of this redbook

1.1 Overview

As e-businesses have evolved to provide content-rich Web sites, a need has arisen to provide the user with a single point of access to content personalized to their interests.

The release of the IBM Commerce Enhancement Pack - April 2003 Edition includes commerce integration code for WebSphere Portal and WebSphere Commerce. This integration functionality is referred to as a commerce enabled portal. When using a commerce enabled portal, users gain the ability to have a single point of access to personalized content and applications in portlets on the portal pages for standard Web browser clients and mobile device clients.

1.1.1 Key concepts of portals

A portal provides a framework for developing portal applications and a runtime environment where portlets can be deployed. The runtime environment is a portal container that runs in a J2EE environment such as the WebSphere Application Server. The portal infrastructure includes the following core set of services:

- ▶ Access to user profile information
- ▶ A framework for portlets to participate in events
- ▶ A framework to communicate with other portlets
- ▶ Access to remote content
- ▶ Access to credentials
- ▶ A framework for storing persistent data

The IBM WebSphere Portal provides the services listed and other important features such as single sign-on, security, Web content publishing, search, personalization, collaboration, enterprise application integration, and support for mobile devices.

This section describes the key concepts associated with portals and portlets, which are the base technology used for commerce enabled portals.

Portal

A portal-enabled Web site typically provides a variety of services such as Web search, news feeds, personalization of content displayed in portlets, e-commerce and links to other sites. The Portal Server provides the base runtime infrastructure and development portlet API for portals. IBM's portal product offering is called IBM WebSphere Portal for Multiplatforms, which comes in three editions (Enable, Extend, and Experience).

Figure 1-1 displays several administration portlets (Welcome, Quick links, World clock, Reminder) displayed on the WebSphere Portal home page.



Figure 1-1 WebSphere Portal home page

Portlet

A portlet is a Java application that is written to the portlet API and hosted on a Portal Server. Portlets can be developed within IBM WebSphere Studio Application Developer using WebSphere Portal Toolkit (plug-in). The portlets are deployed to the WebSphere Portal Server runtime environment. In Figure 1-1, there are several portlets displayed and additional portlets available from the pull-down menu where the Home tab is currently displayed. The portlets displayed and the content within the portlets can be aggregated and personalized.

Figure 1-2 displays the ITSO working example of a commerce enabled portal B2B store. Within this portal page, there are several commerce portlets, including category display, product display and item display.

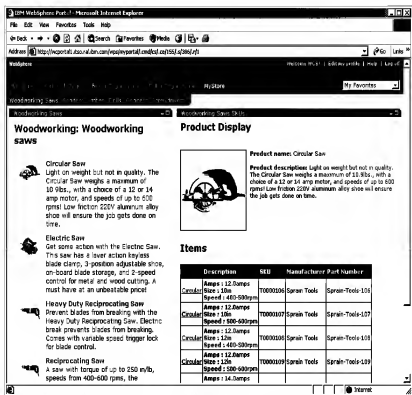


Figure 1-2 WebSphere Commerce Portal page with category, product, and item commerce portlets

Portlet application

A portlet application is a set of portlets grouped together and executable from a single access point. Portlets within the portlet application package share the same context (for example, images, properties files, and classes). The portlet application portlets are packaged into a Web archive (WAR) file. Portlets within the portlet application can communicate with other portlets using custom messages.

Portal page

A portal page displays content and can contain one or more portlets.

Portal place or page group

A portal *place* is a collection of portal pages. The portal administrator can create places, determine which portal pages are in each page, and give the appropriate users authority to access the place and pages.

Portal page layout

The portal page *layout* defines the number of content areas within the page and the portlets displayed within each content area.

Portal permissions

Each portal page is subdivided into one or more content areas. Each content area can contain one or more portlets. The term *permission* refers to the ability of an administrator to control access to users who can edit a page with permissions of move, edit or delete the content areas and the portlets.

Portal themes

The portal *themes* represent the overall look and feel of the portal, including colors, image and fonts.

Portal skins

Portal *skins* refers to the visual appearance of the area surrounding an individual portlet.

Aggregation

Aggregation refers to the process of assembling the contents of each portlet of a portal and rendering the content into one page for the requesting device.

Unlike a traditional Web application where the Web browser client renders the contents of a page received from the server, a portal server aggregates the contents of various portlets on a portal page, renders the page, and then sends the render page to the client.

Personalization

Personalization services are provided using the WebSphere Personalization V4 Rules and Resource Engine (part of WebSphere Portal Content Publisher), which is included with WebSphere Portal V4.2.1 Enable. Personalization is used within the portal to allow for the contents within the portlets to be personalized to the user's interests. Within the context of commerce enabled portals, rule and role-based aggregation is implemented using WebSphere Personalization.

Single sign-on (SSO)

One of the many benefits of a portal Web site is that it can provide a single point of access to content and applications distributed across the enterprise. Single sign-on (SSO) allows the user to move between different applications without being prompted for a user ID and password (or certificate) every time. SSO is especially important for commerce enabled portals, as users will need to move between portlets of different applications without signing on over and over (for example, commerce portlets, portlets external to commerce containing product review info, etc.).

1.1.2 Benefits of a commerce enabled portal

Many businesses strive to improve profitability, reduce cost, and strengthen customer relationships. By interacting more closely with customers and providing them with relevant content, these business goals can be achieved. For example, large enterprise businesses may have several divisions or brands selling products and services on separate e-commerce Web sites. Often opportunities to increase revenues or reduce expenses are not leveraged.

By creating a commerce enabled portal, customers are able to access personalized content and application in portlets of their own choice, or defined by the business hosting the site.

The following are the major reasons why commerce portals are used:

- ▶ Increase productivity and efficiency

Commerce portals provide the means for the user or customer to have a single point of access to information directly related to their needs without hunting through the Web site pages.

The business hosting the commerce enabled portal is provided an infrastructure for managing customers, personalized content, and commerce. In addition, developers are provided common tooling for portal, commerce, and application development.

- ▶ Improve decision making

A portal can put critical information needed for a specific user to make better decisions. For example, a buyer using a portal may have a portlet that provides alerts of products that need to be ordered due to reaching a threshold.

- ▶ Competitive advantage and customer loyalty

Customers appreciate businesses that cater to their needs. By providing a single point of access using a portal, customers' needs and desires for personalized content can provide your business a competitive advantage.

The content displayed in the portlets can be aggregated and personalized based on user role and personalization rules.

- Reach and user experience

Through the use of WebSphere Portal, multiple channels can access your Web site, include PC Web browsers, mobile phones, and wireless PDAs. The WebSphere Portal Server and WebSphere Commerce Server are capable of detecting what type of device is accessing the site and serving the appropriate markup language (HTML, XML, WAP WML) to display content in the portlets.

- Marketing opportunities and increased revenues

A portal provides the environment to personalize content specific to a user's interests. In the case of commerce enabled portals, content displayed in a portlet can be personalized by user role or by personalization rules. For example, content may be personalized to promote the sale of jeans to all customers between 14-21. Personalization and portals provides a means for one-on-one marketing.

Commerce enabled portals also increase opportunities for cross-selling.

Note: For more information on the benefits of commerce enable portals, refer to the following white paper:

- *Pursuing efficiency and revenue with commerce enabled portals* found at:

<http://www.ibm.com/software/webservers/commerce/portal/>

1.1.3 Features of WebSphere Commerce Portal

The commerce enabled portal features included in the IBM Commerce Enhancement Pack - April 2003 Edition (stand-alone or as part of the WebSphere Commerce Portal V5.4) are as follows:

- Base infrastructure
 - Integrated session between WebSphere Portal and WebSphere Commerce
 - Shared user repository through LDAP
 - Single sign-on (SSO)
 - Mobile device browser support (WAP)
 - National language support in commerce enabled portal code
 - Anonymous shopper support
 - Tool to generate commerce portlets
- Dynamic Context Groups

- Portlets can be grouped for content synchronization
- Dynamic Context Groups require no code change in portlets
- Tool to deploy dynamic context portlet groups
- ▶ Real-time communication and collaboration:
 - Integrated WebSphere Commerce messaging subsystem with WebSphere Portal to deliver the following:
 - Alert messaging portlet (MyPortalAlert)
 - Sending SMS messages through WebSphere Commerce to a mobile device
 - Sametime® portlet integration
- ▶ Dynamic rendering of user-relevant content (personalization):
 - Rule-based personalization (for example, customer groups such as teenagers and interests)
 - Role-based personalization (for example, sales manager)
- ▶ WebSphere Commerce portlets and portlet JSPs

Commerce enabled portlets and portlet JSPs have been developed for the business-to-consumer (B2C) WebFashion sample store template and business-to-business (B2B) ToolTech sample store template.

Some key portlets that have been developed are as follows:

- Shopping flow support
 - Product catalog portlet
 - Product search portlet
 - New arrivals portlet
 - Shopping cart portlet
 - Order status portlet
- Customer self-help
 - Order status portlet
 - User account portlet
- Marketing e-spot
- ▶ WebSphere Commerce Portal Tooling Framework

The WebSphere Commerce Portal Tooling Framework is implemented by installing a WebSphere Studio Application Developer plug-in. The WebSphere Studio Application Developer can be used to generate new WebSphere Commerce Portal portlets and create or modify portlet JSPs.

1.1.4 WebSphere Commerce Portal component packaging

The integration of WebSphere Commerce V5.4.0.5 and WebSphere Portal V4.2.1 can be achieved by obtaining separate software components or by purchasing the IBM WebSphere Commerce Portal V5.4 product offering.

The following high-level components are needed to achieve the desired WebSphere Commerce V5.4 and WebSphere Portal V4.1.x integration:

- ▶ IBM WebSphere Commerce V5.4
- ▶ IBM WebSphere Commerce FixPak V5.4.0.5
- ▶ IBM Commerce Enhancement Pack - April 2003 Edition and APAR JR18068
- ▶ IBM WebSphere Portal V4.2.1 Enable

IBM WebSphere Portal packaging

At the time of writing this redbook, IBM offered three editions of WebSphere Portal, Enable, Extend and Experience. While developing solutions for this redbook, we used WebSphere Portal V4.2.1 for Multiplatforms.

For additional information on WebSphere Portal for Multiplatforms, refer to the following URL:

<http://www.ibm.com/software/webservers/portal/>

Commerce enabled portal component based offering

A component based approach to assembling the necessary software for a commerce enabled portal is suited to customers who have already purchased IBM WebSphere Commerce V5.4 (Business or Professional Edition) and would like to add commerce enabled portal functionality. In this case, the customer can purchase WebSphere Portal V4.2.1 Enable and download the following to enhance WebSphere Commerce V5.4:

- ▶ IBM WebSphere Commerce FixPak V5.4.0.5
- ▶ IBM Commerce Enhancement Pack - April 2003 Edition

The Commerce Enhancement Pack contains the commerce enabled portal runtime and development components.

- ▶ WebSphere Commerce Portal APAR JR18068

This APAR includes fixes to the commerce enabled portal code and was made available after the IBM Commerce Enhancement Pack - April 2003 Edition.

IBM WebSphere Commerce Portal V5.4 product offering

This product offering is called IBM WebSphere Commerce Portal V5.4 and it includes the following:

- ▶ IBM WebSphere Commerce V5.4, Professional Edition
- ▶ IBM WebSphere Commerce FixPak V5.4.0.5
- ▶ IBM Commerce Enhancement Pack - April 2003 Edition
- ▶ IBM WebSphere Portal Enable V4.2

For more information on the IBM WebSphere Commerce Portal V5.4 product offering refer to:

<http://www.ibm.com/software/webervers/commerce/portal/>

Note: The IBM WebSphere Commerce Portal V5.4 includes the WebSphere Commerce Professional Edition. If B2B functionality is needed, you will need to assemble the commerce enabled portal components as described.

1.1.5 Prerequisite knowledge and information sources

The information found in this redbook is intended for IT architects, specialists and developers who need to integrate the WebSphere Commerce and WebSphere Portal. The WebSphere Commerce Portal architecture is built upon the technologies of WebSphere Commerce and WebSphere Portal. For this reason, it is important that the reader have a firm understanding of the WebSphere Commerce and WebSphere Portal architecture. We have listed key sources of information for WebSphere Commerce, WebSphere Commerce Portal, and WebSphere Portal.

WebSphere Commerce information

Several sources information can be found for IBM WebSphere Commerce V5.4 relevant to commerce enabled portal architecture, design and implementation:

- ▶ The following IBM WebSphere Commerce V5.4 product guides and online information are included with the product or can be downloaded:

Business Edition:

http://www.ibm.com/software/webervers/commerce/wc_be/lit-tech-general.html

Professional Edition:

http://www.ibm.com/software/webervers/commerce/wc_pe/lit-tech-general.html

- *Fundamentals Guide, IBM WebSphere Commerce V5.4*

This product guide provides and overview of the WebSphere Commerce architecture and features.

- *Programmer's Guide, IBM WebSphere Commerce V5.4*

This product guide provides programming reference information for developing WebSphere Commerce store-front and back-end assets. Included in this guide is a description of the WebSphere Commerce programming model and architecture.

- *Store Developer's Guide, IBM WebSphere Commerce V5.4*

This product guide describes the architecture of the store data, database schema and objects. The guide provides detailed information on how to manage such data assets as products, categories, taxes, and shipping.

- *WebSphere Commerce V5.4 online documentation*

- Depending on the desired operating system platform, select the appropriate WebSphere Commerce installation guide. For example, on Windows®:

- *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Windows*
- *Additional Software Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Windows*
- *Installation Guide, IBM WebSphere Commerce FixPak V5.4.0.5*

- *Installation Guide, IBM WebSphere Commerce Studio V5.4 for Windows NT and Windows 2000*

This guide provides some basic procedures for installing and configuring the WebSphere Commerce development environment.

► **IBM Redbooks**

- *WebSphere Commerce V5.4 Handbook, SG24-6567*

This redbook includes architecture guidelines for security, scalability, performance tuning, and testing. In addition, the redbook includes many advanced working example implementation and integration scenarios.

- *Mobile Commerce Solutions Guide, using WebSphere Commerce Suite V5.1, SG24-6171*

This redbook provides detailed architecture information and working examples for the WebSphere Commerce mobile architecture, which is used in the integration of WebSphere Commerce and WebSphere Portal.

- *WebSphere Commerce V5.4 Catalog Design and Content Management, SG24-6585*

This redbook describes the store and data architecture, and also provides a working example for managing data using Catalog Manager.

- *B2B e-commerce With WebSphere Commerce Business Edition V5.4, Patterns for e-business Series*, SG24-6194

This redbook includes Business patterns, Composite patterns, Application patterns, Runtime patterns and Product mappings based on the IBM Patterns for e-business.

WebSphere Commerce Portal information

To find more information on WebSphere Commerce Portal, refer to the following:

- *Getting Started, IBM Commerce Enhancement Pack*
This guide provides procedures for installing and configuring the IBM Commerce Enhancement Pack - April 2003 Edition, of which WebSphere Commerce Portal is a part.
- *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*
We used this guide when writing Chapter 5, "Implement the runtime environment" on page 165, which provides a procedure for installing and configuring an advanced commerce enabled portal runtime environment.

WebSphere Portal information

For WebSphere Portal development and the portlet API information, refer to the following:

- *Portlet Developer's Guide, WebSphere Portal V4.1*
- *WebSphere Portal V4.1 Developer's Handbook*, SG24-6897
- *WebSphere Portal V4.1 Handbook Volume 2*, SG24-6920 redbook
- *Mobile Applications with IBM WebSphere Everyplace Access Design and Development*, SG24-6259

For WebSphere Portal runtime environment information, refer to the following:

- *WebSphere Portal V4.1 Handbook Volume 1*, SG24-6883 redbook
- *WebSphere Portal V4.1 Handbook Volume 3*, SG24-6921 redbook
- *WebSphere Portal V4.1, Windows 2000 Installation*, REDP3593
- *Access Integration Pattern using IBM WebSphere Portal Server*, SG24-6267
- *A Portal Composite Pattern Using WebSphere Portal V4.1.2*, SG24-6869
- *WebSphere Portal InfoCenter:*
<http://publib.boulder.ibm.com/pvc/wp/42/smb/en/InfoCenter/index.html>

1.2 Structure of this redbook

This redbook is organized into three parts and designed to serve different audiences such as IT architects, IT specialists, developers, sales professionals, and business users.

Part 1: Introduction to IBM WebSphere Commerce Portal

Part 1 of this redbook is intended for sales professionals and business users as well as developers, IT architects, and IT specialists. The chapters in this part provide a foundation of the key concepts and overview on such topics as the features and architecture of WebSphere Commerce Portal.

The chapters in Part 1 of this redbook are as follows:

- ▶ Chapter 1, "Introduction to commerce enabled portals" on page 3
- ▶ Chapter 2, "WebSphere Commerce Portal architecture" on page 17
- ▶ Chapter 3, "Patterns for e-business and commerce enabled portals" on page 63

Part 2: Working example scenario: commerce enabled portal

Part 2 of this redbook is intended for sales professionals and business users as well as developers, IT architects, and IT specialists. The chapters in this part provide an end-to-end working example.

The chapters in Part 2 of this redbook are as follows:

- ▶ Chapter 4, "Business requirements analysis and solution design" on page 99
- ▶ Chapter 5, "Implement the runtime environment" on page 165
- ▶ Chapter 6, "Implement the development environment" on page 253
- ▶ Chapter 7, "Create a commerce enabled portal store" on page 285
- ▶ Chapter 9, "Deploy the commerce enabled portal" on page 309
- ▶ Chapter 8, "Create portlets with Dynamic Context Groups" on page 301
- ▶ Chapter 10, "Creating a commerce enabled portal for mobile clients" on page 347

Part 3: Appendixes

Part 3 of this redbook includes the following:

- ▶ Appendix A, "WebSphere Portal and WebSphere Commerce sample LDIF files" on page 357
- ▶ Appendix B, "Logon for commerce portlets without single sign-on" on page 361
- ▶ Appendix C, "Tips and troubleshooting for commerce enabled portals" on page 369
- ▶ Appendix D, "Additional material" on page 377



WebSphere Commerce Portal architecture

This chapter describes several approaches for implementing a commerce enabled portal solution, and explains the architectural decisions behind the integration approach selected for the IBM WebSphere Commerce Portal. In addition, we describe the following architecture elements of the WebSphere Commerce Portal: runtime environment, component architecture, development environment, and HTTP request flow from a Web browser and mobile client.

We include a review of the WebSphere Commerce and WebSphere Portal architecture to serve as a baseline in our comparison with WebSphere Commerce Portal. The comparison includes the following features of the solution: mobile client access, authentication and single sign-on, access control, session management, presentation services and page rendering.

The chapter is organized into the following section:

- ▶ WebSphere Commerce Portal architecture
- ▶ WebSphere Commerce and WebSphere Portal architecture overview
- ▶ Mobile client access
- ▶ Authentication and single sign-on
- ▶ Access control
- ▶ Session management
- ▶ Presentation services and page rendering

2.1 WebSphere Commerce Portal architecture

This section describes the following in the WebSphere Commerce Portal architecture:

- ▶ Commerce enabled portal integration approaches
- ▶ XML-based commerce enabled portal solution outlook
- ▶ WebSphere Commerce Portal runtime environment
- ▶ WebSphere Commerce Portal component architecture
- ▶ WebSphere Commerce Portal HTTP request flow
- ▶ WebSphere Commerce Portal architecture of key features

2.1.1 Commerce enabled portal integration approaches

There are several ways in which a commerce enabled portal can be implemented. This section describes four commerce enabled portal integration approaches that were evaluated and explains the architectural decisions behind the architecture approach used by the WebSphere Commerce Portal. It is important to note that the selection of a commerce enabled portal solution may vary depending on the integration objectives and requirements of a business.

For example, the approach selected can vary based on requirements such as the following:

- ▶ Add WebSphere Commerce Portal to an existing WebSphere Commerce site.
In this case, an existing WebSphere Commerce site will need to be extended with portal functionality. A simple and quick method of converting existing standard WebSphere Commerce JSPs to WebSphere Commerce Portal portlet JSPs may drive the architecture decision.
- ▶ Add WebSphere Commerce Portal functionality to an existing WebSphere Portal site.
In this case, there may be large numbers of existing standard WebSphere Portal portlets and portlet JSPs. A solution that allows for extending the existing standard WebSphere Portal portlet investment may drive the architecture decision.

Approach 1: Portlet through servlet chaining

This approach uses the servlet chaining capability provided by the WebSphere Portal ServletInvokerPortlet. The ServletInvokerPortlet calls a RequestDispatcher servlet which enables a servlet or JSP to process a request partially and then invoke another servlet, JSP or HTML page to do further processing. This functionality is provided in the Servlet API in the form of the `javax.servlet.RequestDispatcher` interface. This is implemented by casting the PortletSession into the `HttpServletSession`. This approach can apply to

integration of any typical Web application that has a RequestDispatcher servlet or JSP components that are accessible as URLs.

When a PortletRequest is made to the portlet, the PortletRequest is cast into a HttpServletRequest for the Web application (for example, WebSphere Commerce). WebSphere Commerce (or Web application) returns a HttpServletResponse, which is then cast back into the calling portlet as PortletResponse.

Approach 2: Native WebSphere Portal portlet front-end

This integration approach results in a more fundamental change in a WebSphere Commerce Web application. Integration approach 2 envisions the portlet being the native client front-end of the Web application. The portlets communicate to the Web application's business logic components and resources directly through PortletRequest and PortletResponse.

The Web application interfaces directly and natively with the PortletRequest and PortletResponse. The implication is that WebSphere Portal is the only presentation platform. This approach results in fundamental changes to the Web application and, therefore, should be considered carefully.

Approach 3: Portlet as HTTP proxy to WebSphere Commerce

This integration approach uses a WebSphere Commerce portlet to generate new HTTP requests to the WebSphere Commerce Web pages. In other words, the WebSphere Commerce portlet functions as the WebSphere Commerce HTTP proxy, routing requests and responses to another Web server.

Approach 4: Portlet as application launch pad

In this approach, the portlet launches the Web application through a URL link into a separate Web browser window. For example, <http://www.ibm.com/shop> is being launched as a URL link from the WebSphere Commerce portlet. Once the URL is launched, the user is no longer participating with the portal.

WebSphere Commerce Portal architecture decisions

All four above-mentioned integration approaches were evaluated when selecting the WebSphere Commerce Portal solution. During the evaluation, the following criteria were used in the selection of the commerce enabled portal approach:

- ▶ Solution provides seamless integration for the user between WebSphere Portal and WebSphere Commerce
- ▶ Solution is capable of being implemented quickly
- ▶ Solution does not require modifying the base WebSphere Commerce architecture

Approach 4 was not considered in any great detail because it did not meet the integration goal and requirement of being seamless to the user. Approach 2 was not selected because it would require significant technical resources and time to convert the assets of an existing WebSphere Commerce store.

Initially, approach 1 (servlet chaining through ServletInvokerPortlet from WebSphere Portal) was the choice of implementation approach, because this fits very well into the WebSphere Commerce architecture, where WebSphere Commerce capabilities are accessible as servlets. The major attraction is the fact that it does not require major architectural changes to WebSphere Commerce.

However, one major drawback of approach 1 lies in the limitation of servlets not able to chain to another servlet residing in another Web container. As the ServletContext object is not remoteable by nature, the RequestDispatcher of the distributed servlet cannot be obtained. This means the ServletInvokerPortlet cannot chain servlets that reside in another Web container.

Chaining WebSphere Commerce servlets is a clean way of integrating, which fits the WebSphere Commerce architecture very well and is the model of choice. To support a more distributed topology of integration, however, the ability to include remote servlets should be added to approach 1.

The decision is to provide the remote servlet call services in the WebSphere Commerce portlets (and WebSphere Commerce portlet application) that *chain* WebSphere Commerce servlets residing remotely. The WebSphere Commerce servlets are being called through new HTTP requests made by the WebSphere Commerce portlets. These new HTTP requests maintain a one-to-one mapping of the PortletRequests of the calling WebSphere Commerce portlet application, relaying critical session information (such as cookies including an LtpaToken, session context and so forth) from the original WebSphere Commerce portlet requests to the new HTTP requests that call the remote WebSphere Commerce servlets.

WebSphere Commerce Portal architecture approach selection

The WebSphere Commerce Portal integration approach selected and implemented is a modified approach 1 (chaining servlet) with the remote servlets (WebSphere Commerce commands) performed through a modified approach 3. In this approach, the WebSphere Commerce portlets issue HTTP calls to remote WebSphere Commerce command servlets, relaying information and data from the original portlet requests to the new HTTP WebSphere Commerce servlet calls.

2.1.2 XML-based commerce enabled portal solution outlook

When writing this redbook, we investigated using an XML-based approach consistent with the Model-View-Controller (MVC) application architecture for commerce enabled portals. This type of solution is more consistent with native portal development described in "Approach 2: Native WebSphere Portal front-end" on page 19. Due to time constraints, we were not able to fully implement this type of solution, as it is significantly different from what is provided by the WebSphere Commerce Portal solution and requires a fundamental architecture change. This section attempts to highlight the benefits and high-level architecture of an XML-based MVC-type commerce enabled portal solution.

The Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. A commerce enabled portal application could potentially be built to leverage this XML-based approach for the following reasons:

- Render the markup language for the given client using XML data and XSLT stylesheets.
- Support the Model-View-Controller aspect by separating the business logic, data objects, and their rendering.
- Easily access and integrate back-end systems including powerful transformations.

Figure 2-1 provides an overview of how XML can simplify the development and the integration of commerce enabled portals.

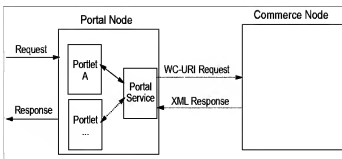


Figure 2-1 Commerce enabled portals leveraging XML

To generate XML instead of HTML or WML, commerce portlet JSPs would need to be modified. The customization includes the creation of the XML schema

definition and generation of the XML structure. This includes the data elements as a result of a WebSphere Commerce URI request.

The portal node receives the requests from a client and delivers the HTML or WML markup language as requested. Within the portal server, a portlet has the responsibility of rendering the content markup and to interact with the back-end system. Using XML with the commerce enabled portal architecture would support the following:

1. Portlets are interacting natively with the commerce node. They expect that the response contains XML, which is rendered by applying a stylesheet or programming the logic within the portlet.
2. A portal service is established that handles the portlet request. The service covers the interaction with the commerce node and provides a service level interface for portlets. It maintains the necessary information to contact the commerce node. The XML data stream is passed to the calling portlet. The portlets can interact with the commerce node seamlessly while rendering the XML response from the back-end for the client markup.

The major advantage of using XML is the clear separation of back-end services such as the commerce node and the rendering engine such as the portal node. In a wider scope, requests could be tunneled using a message oriented approach. Moreover, the response could be enriched and transformed using sophisticated broker and transformation capabilities.

2.1.3 WebSphere Commerce Portal runtime environment

The WebSphere Commerce Portal runtime environment displayed in Figure 2-2 on page 23 represents the nodes of the commerce enabled portal solution grouped as functional and operational components. The objective of this section is to highlight the key nodes (numbered nodes) used in the integration solution in support of understanding the lower-level components architecture of WebSphere Commerce Portal.

Note: For a much closer look at Application patterns, Runtime patterns and Product mappings (details on product versions used in the ITSO test environment), refer to Chapter 3, "Patterns for e-business and commerce enabled portals" on page 63.

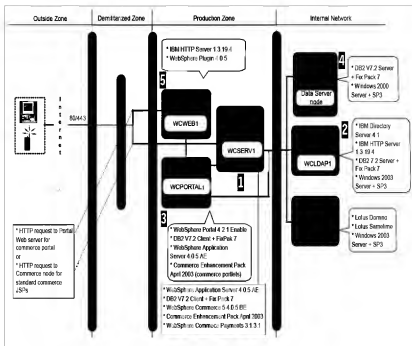


Figure 2-2 WebSphere Commerce Portal runtime environment nodes

Note: Runtime and development environment implementation

The implementation procedure for the runtime environment displayed in Figure 2-2 on page 23 is detailed in Chapter 5, "Implement the runtime environment" on page 165.

The implementation procedure for the development environment used by the ITSO is detailed in Chapter 6, "Implement the development environment" on page 253.

2.1.4 WebSphere Commerce Portal component architecture

This section describes the component architecture of the WebSphere Commerce Portal in relation to the key features mentioned in 1.1.3, "Features of WebSphere Commerce Portal" on page 9. We will start with the high-level component architecture and drill down to specific components and features provided.

WebSphere Commerce Portal component architecture

The component architecture of the WebSphere Commerce Portal solution is largely a combination of WebSphere Portal and WebSphere Commerce components, as seen in Figure 2-3. The key integration code provided with WebSphere Commerce Portal is contained within the WebSphere Commerce portlets.

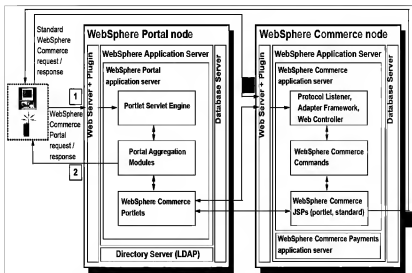


Figure 2-3 WebSphere Commerce Portal component architecture

As displayed in Figure 2-3, standard Web browser clients (and mobile clients with additional runtime components implemented) can access the WebSphere Commerce node directly (see 1 in Figure 2-3). When using a commerce enabled portal solution, client requests are directed to the WebSphere Portal node where commerce portlets then interact with the WebSphere Commerce node.

When WebSphere Commerce integrates with a WebSphere Portal, the commerce portlets running on the portal server initiate multiple new and independent HTTP requests to the WebSphere Commerce Server to perform commerce business processes. From the perspective of the WebSphere Commerce Server, the commerce portlets function as an alternate User Agent client using the WebSphere Commerce adapter technology. Responses from WebSphere Commerce are being rendered as HTTP responses to the corresponding HTTP requests of the commerce portlets by the WebSphere Portal Aggregation Modules.

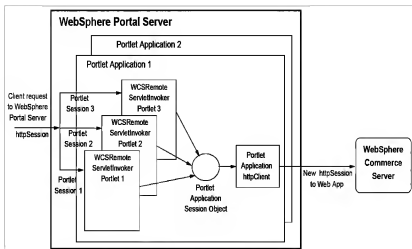


Figure 2-4 WebSphere Commerce Portal high-level integration architecture

Multiple HTTP requests and HTTP responses between commerce portlets/commerce portlet JSPs and WebSphere Commerce within the same portal user session must be perceived by WebSphere Commerce as one cohesive commerce user session, as seen in Figure 2-4.

Figure 2-5 on page 26 displays a lower-level architecture view of the integration for multiple HTTP requests and HTTP responses between commerce portlets/commerce portlet JSPs and WebSphere Commerce within the same portal user session.

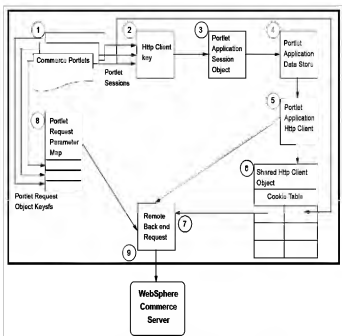


Figure 2-5 WebSphere Commerce Portal low-level integration architecture

2.1.5 WebSphere Commerce Portal HTTP request flow

This section describes the high-level flow of an HTTP request from a Web browser client to a WebSphere Commerce Portal Web site, which is depicted in Figure 2-6 on page 27. Let's say for example that a customer browsing your commerce enabled portal Web site wishes to view a product category in the online catalog. One might consider the interaction of components to fulfill this request.

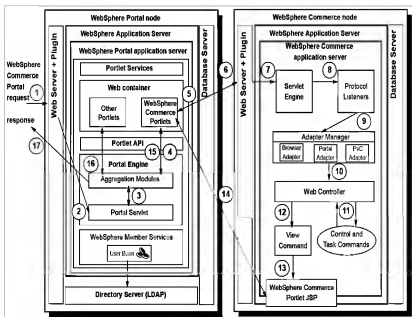


Figure 2-6 WebSphere Commerce Portal HTTP request flow

The high-level flow of the HTTP request to perform the portlet product category display as depicted in Figure 2-6 is as follows:

1. The request to view the category portlet is sent from the Web browser client to the Web server of the WebSphere Portal node.
2. The Web server accepts the request, and then uses the WebSphere Application Server plug-in to direct the request to the Portal Engine, more specifically the Portal Servlet.
3. The Portal Servlet interacts with the Portal Aggregation Module.
4. The Portal Aggregation Module is responsible for aggregating content from various portlets for a target device, and then rendering the contents of the portlets into one portal page. This step involves the aggregation of content by calling the portlets to perform specific tasks.
5. In our example, the user requested a category portlet. The Aggregation Module has requested content to be retrieved by the commerce portlet. The commerce portlet will then interact with WebSphere Commerce by using the WCSRremoteServletInvoker (see Figure 2-4 on page 25 and Figure 2-5 on page 26).

6. The commerce portlet WCSRRemoteServletInvoker sends a request to the Web server on the WebSphere Commerce node.
7. The Web server accepts the request, and then uses the WebSphere Application Server plug-in to direct the request to the WebSphere Commerce Servlet Engine.
8. The WebSphere Commerce Servlet Engine assigns the request to its own thread, and dispatches the request to the correct protocol listener.
9. The protocol listener in this example is the HTTP request servlet. The request servlet passes the request on to the Adapter Manager.

The Adapter Manager determines that the request came from a standard Web browser client, and invokes the appropriate adapter. In the case of a request from the WebSphere Portal node (WCSRRemoteServletInvoker portlet), this is the Portal Adapter. For standard Web browser clients accessing the WebSphere Commerce node directly, this is the HTTP Browser Adapter. For mobile clients, it is the PvC Adapter (for example WAP Adapter).

10. The adapter performs little function in the case of an HTTP request from a PC browser. Adapters can be written to handle the more complex cases. The HTTP browser adapter passes the request on to the Web Controller.

In the case of the Portal and PvC Adapter, they are used to determine the requesting device type by analyzing the User Agent of the HTTP request, and setting the root directory of JSPs to eventually be served to the client (or portal aggregation module).

11. The Web Controller determines the correct command to invoke based on the uniform resource identifier (URI) by querying the WebSphere Commerce database. In this example, the customer requested a category display, so the category display command is invoked. The command performs some processing and then returns a view name to the Web Controller.
12. The Web Controller determines the correct JSP to display for the view and the required parameters, by searching the database VIEWREG table.
13. The JSP writes a response, which is returned to the Web server.

Commerce portlet JSPs do not contain HTML tags, such as <header> or <html>. The portal aggregation module will eventually assemble and render the contents to create a portal page to be displayed.

14. The Web server in turn relays that response to the client for the display of the category display page. In the case of a portal, the Portal Engine in a sense is the client.

The commerce portlet JSP is returned to the commerce portlet.

15. The commerce portlet returns the content of the commerce portlet JSP to the Portal Aggregation Module.

16. The Portal Aggregation Module aggregates the content of the portlet JSPs received from the individual portlets and renders the portal page.
17. The portal page containing the product category display is returned to the client to be displayed in the Web browser.

2.1.6 WebSphere Commerce Portal architecture of key features

This section describes the high-level architecture of the following key features of WebSphere Commerce Portal:

- ▶ Dynamic Context Groups
- ▶ Rule and role-based dynamic aggregation
- ▶ Real-time communication and collaboration

Dynamic Context Groups

Portlets are by nature independent of one another. WebSphere Portal provides the portlet messaging capability such that portlets within the same portlet application can communicate with one another. WebSphere Commerce Portal extends the portlet messaging capability to achieve dynamic business content synchronization. The design of dynamic context portlet grouping provides a facility for a user to group portlets whose content is meant to be synchronized dynamically at runtime. Such a facility does not require code changes for the participating portlets within a Dynamic Context Group, but instead, Dynamic Context Grouping is done through the deployment descriptor. This greatly enhances the ease of regrouping of portlets without the need to restart the portal server as part of the regrouping exercise.

Note: For more detailed information on Dynamic Context Groups, refer to the following:

- ▶ *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*
- ▶ Chapter 8, "Create portlets with Dynamic Context Groups" on page 301

Applying Dynamic Context Groups

Dynamic business content synchronization is needed when the content rendered in a given portlet is a direct function of the latest content of other portlets. Consider the following scenario. A ViewAccountList portlet displays the list of accounts that a sales manager is responsible for managing. When the sales

manager selects a specific account from the account list, the following actions take place dynamically without user interface being required:

- ▶ `ViewInvoiceList` portlet displays the list of outstanding invoices of the selected account
- ▶ `ViewOrderlist` portlet displays the order list of the selected account
- ▶ `ViewAccountDetailedInformation` portlet displays the account detail information of the account selected.

When the sales manager selects a different account from the account list, all the related portlets automatically get refreshed to show synchronized information.

This kind of portlet integration and synchronization “at the glass” is often effective and lightweight, especially when no back-end data store update is involved in the integration. In this case, the portlet is just performing a different query and display based on the change of context, as the above example scenario illustrates. Integration resulting in a sequence of back-end changes is better accomplished through other integration means at the back-end level such as Web services.

How Dynamic Context Groups work

A given Dynamic Context Group has the following elements:

- ▶ Dynamic context

Dynamic context is the context that all portlets within the same Dynamic Context Group share. There should only be one dynamic context per Dynamic Context Group. In the above example scenario, the `AccountID` selected by the sales manager from an account list is the dynamic context.

- ▶ Dynamic context master portlet

There should only be one dynamic context master portlet per Dynamic Context Group. The dynamic context master portlet is responsible for sending portlet messages to the dynamic context slave portlets when the dynamic context is set or changed. In the previous example, the `ViewAccountList` portlet is the master portlet. When a sales manager selects a specific account, the dynamic context is set to the selected account's `AccountID`. The dynamic context master portlet picks up the setting of the dynamic context and sends the portlet message to the dynamic context slave portlets.

Each dynamic context master portlet has a built-in `ActionListener`. The `ActionListener` listens for the dynamic context (as defined in the Dynamic Context Group) in the user actions of the master portlet. When the predefined context is detected in the user action, the dynamic context master portlet generates the event that send portlet messages to the dynamic context slave portlets.

► Dynamic context slave portlets

Each Dynamic Context Group must have at least one dynamic context slave portlet. Each group can have more than one dynamic context slave portlets. Each dynamic context slave portlet has a predefined slave action for the slave. The slave portlet, upon receiving its master's portlet messages, will perform pre-registered slave action according.

In the previous example, the dynamic context slave portlets are `ViewInvoiceList`, `ViewOrderList`, and `ViewAccountDetailedInformation`. The registered slave action for the `ViewInvoiceList` portlet is the URI to fetch the `InvoiceList` with the account instance as the dynamic context.

Each dynamic context slave portlet has a built-in portlet message listener that listens to the message event generated by the dynamic context master portlet. When the notification of the master portlet is received, the slave portlet performs the pre-registered slave action, creating the effect of synchronized content based on the current context.

From the portlet point of view, a given portlet can be a member of more than one Dynamic Context Group, playing different roles in each Dynamic Context Group. For example, a given portlet can be the master in one group but slave in another group, as Figure 2-7 on page 32 and Table 2-1 on page 32 illustrate.

At portlet application deployment time, existing portlets can form Dynamic Context Groups. First, users create the dynamic context portlet group definition in a Dynamic Context Group XML instance that articulates:

- The Dynamic Context Group name
- The dynamic context for that group
- The dynamic context master portlet
- The list of dynamic context slave portlet(s)
- The action definition of each slave

A dynamic context portlet XSD file is used for validation of the dynamic group defined.

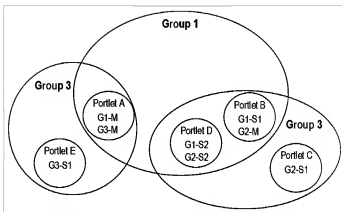


Figure 2-7 WebSphere Commerce Portal: Example of Dynamic Context Groups

Table 2-1 Example of Dynamic Context Groups

Group	Master/Slave	Group ID	Portlet
Dynamic Group 1			
	Master Portlet	G1-M	Portlet A
	Slave Portlet	G1-S1	Portlet B (1st priority)
	Slave Portlet	G1-S2	Portlet D (2nd priority)
Dynamic Group 2:			
	Master Portlet	G2-M	Portlet B
	Slave Portlet	G2-S1	Portlet C (1st priority)
	Slave Portlet	G2-S2	Portlet D (2nd priority)
Dynamic Group 3:			
	Master Portlet	G3-M	Portlet A
	Slave Portlet	G3-S1	Portlet E

Then users use the Dynamic Context Portlet Group Assembly Tool (a WebSphere Studio Application Developer plug-in) to:

- Validate the dynamic context portlet group xml instance previously defined with the dynamic context portlet group XSD file.

- Update the `portlet.xml` deployment descriptor of each portlets to reflect the Dynamic Context Group mapping as defined in the Dynamic Context Group XML instance.

After the `portlet.xml` file is updated, it is repackaged in its portlet application WAR file ready for deployment.

At runtime, the user interaction with the dynamic context master portlet results in the change of the dynamic context as defined by the Dynamic Context Group. The master portlet's built-in `DynamicContextActionListener` detects the user's action and notifies all the slave portlets within the same Dynamic Context Group. The built-in `DynamicContextMessageListener` of the slave portlet receives the notification. Each of the slave portlets performs the slave action as defined in the Dynamic Context Group instance definition. As a result, content of different portlets get synchronized.

Rule and role-based dynamic aggregation

The ability to aggregate portal resources that are relevant to the user who logs on to WebSphere Commerce Portal is critical to the effectiveness of the business goal of the site. It is the basis upon which customer loyalty is built. The WebSphere Commerce Portal can enable site owner to define business rules mapping to a defined set of portal resources. Portal users who fit a certain classification of a given set of business rules will see a relevant set of portal resources being dynamically rendered to them when the user logs on.

Note: For more detailed information on rule and role-based aggregation, refer to the following:

- *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*
- *WebSphere Portal V4.1 Developer's Handbook*, SG24-6897 in the chapter on WebSphere Personalization

Rule-based dynamic aggregation

Consider the following scenario. A clothing retail store wants to build a user-relevant site selling their brand of clothes. The business strategy of this online clothing retail store this month is to sell via targeted age groups. To this end, the following steps are required:

1. Set up the classification definitions for teenagers, young adults, middle-aged adults, and seniors, etc.
2. Portal resources are being grouped into content groups per each previously defined classification in step 1. For example, the teenager's content group

includes jeans and T-shirt promotion portlet, teen's new arrival portlet, and other teenager-related portal resources.

At runtime, when the user logs on, the Personalization Engine evaluates the user's information in the user registry against all defined classifications stored in the rules database. A given user may fit multiple classifications. For example, a user who is 16 years old fits the classification of a teenager: female-gender group, electronic-interest group. For all classifications that a given user falls within, the portal resources defined in each content group set for each classification will be sent to the WebSphere Portal Aggregation Module to aggregate.

The clothing retail store can change their business strategy from selling by age group to selling by interest. Instead of re-programming the entire site, the support for this change in business strategy can be done using the following steps:

- ▶ Defining new rules for each classification of user interests in the rules database
- ▶ Defining portal resources for each new content groups and for each interest classification

WebSphere Commerce Portal makes possible the increase in productivity and flexibility to support these kinds of business requirement changes. These business requirement changes can be implemented using deployment and configuration steps instead of a complete code rewrite or implementation for the site.

Rules were defined using the Personalization wizard and were captured in the rules database. Content groups per classification are previously defined with a set of portal resources for a defined classification. These content groups of portal resources are also personalization resources.

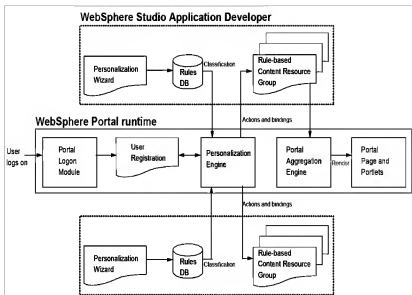


Figure 2-8 Rule and role-based dynamic aggregation

Role-based dynamic aggregation

Role-based definitions can be treated as a kind of rule-based classification (see Figure 2-8). Each business role of a given organization has a defined set of tasks that comes with the business role. This defines the portal resources that a given business role needs to see when a user of this business role logs on.

In WebSphere Commerce Portal, each classification of business roles is stored in the roles database. In fact, the roles database is actually a rules database. The roles database holds the organization definition of business role for the portal server. Portal resources are being grouped into content resource group per each business role.

At runtime, when a user of a given business role logs on, the Personalization Engine evaluates the user role as stored in the user registry against the rules database that contains the rules definition of each role. All portal resources defined per each content group for all matched roles classification will be sent to the portal aggregation module for rendering.

Real-time communication and collaboration

One of the key benefits of the WebSphere Commerce Portal is the out-of-the-box communication facilities. The WebSphere Commerce Portal integrates

WebSphere Portal with the WebSphere Commerce common messaging subsystem to provide additional communication vehicles via WebSphere Commerce Portal, including the following:

- ▶ Alert messaging portlet (MyPortalAlert)
- ▶ Short Message Services (SMS) support for mobile devices
- ▶ Enhanced and integrated Sametime portlets

Note: For more detailed information on the real-time communication and collaboration features of WebSphere Commerce Portal, refer to the *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*.

Alert messaging portlet (MyPortalAlert) and SMS

The alert messaging portlet (MyPortalAlert), shipped as a part of the WebSphere Commerce Portal, displays text messages originated by WebSphere Commerce subsystems or as originated by an administrator. Users of WebSphere Commerce Portal can also use the alert messaging portlet to subscribe message groups in the commerce enabled portal.

Each WebSphere Commerce business subsystems such as order and campaign can send messages to either an individual user or a group of users. These messages can be delivered to the Alert messaging portlet or to a Short Message Services (SMS) mobile device, or to both. Users of commerce enabled portals can subscribe to a delivery channel through WebSphere Commerce Portal.

When an administrator or the commerce subsystem sends a message to a given user or a group of users using the WebSphere Commerce Messaging subsystem, the message is inserted in the message database. When each user logs on, the MyAlertPortal displays the user messages directed to them by calling a WebSphere Commerce command to retrieve all relevant messages. All retrieved messages will be sent as responses being rendered inside the MyPortalAlert.

Each user can subscribe to message groups of interest and decide on the message delivery channel choices (SMS, MyPortalAlert or both).

For SMS messages, the WebSphere Commerce Messaging subsystem transfers the SMS messages to the WebSphere Everyplace® Wireless Gateway (EWG) using the gateway's Push Toolkit APIs.

Sametime integration

To better integrate Sametime capability into WebSphere Commerce Portal, a specialized Sametime portlet has been built as a tactical solution. This specialized Sametime portlet has a built-in Sametime HTML controller with a view JSP. The Sametime HTML controller retrieves the user credentials from the

portlet session. The view JSP downloads the Sametime applet from the Sametime server, passing the user credential, the server name, and the server port number to the Sametime applet. The Sametime applet then renders within the Sametime portlet communicating with the Sametime server, achieving the effect of native integration of Sametime capabilities in the WebSphere Commerce Portal.

2.2 WebSphere Commerce and WebSphere Portal architecture overview

When designing a commerce enabled portal solution, it is important to understand the existing architecture of a deployed site, including the base architecture of WebSphere Commerce and WebSphere Portal. Architectural decisions are often influenced by existing technologies. This section provides an overview of the WebSphere Commerce and WebSphere Portal architecture.

2.2.1 WebSphere Commerce architecture

WebSphere Commerce is largely based on the Java 2 Enterprise Edition (J2EE) architecture found in the WebSphere Application Server. For this reason, the WebSphere Commerce architecture shares many benefits of the WebSphere architecture for scalability, security, and flexible programming environment.

We will focus our attention on the following aspects of the WebSphere Commerce architecture:

- ▶ WebSphere Commerce runtime environment
- ▶ WebSphere Commerce subsystems and components
- ▶ WebSphere Commerce HTTP request flow

WebSphere Commerce runtime environment

WebSphere Commerce can be implemented in various runtime configurations such as single-tier, two-tier, three-tier and three-tier enterprise. For the purposes of this chapter, we have highlighted the nodes seen in Figure 2-9 on page 38 that are relevant to a comparison of WebSphere Portal and WebSphere Commerce Portal. For a standard WebSphere Commerce site implementation, the Directory (LDAP) node and Collaboration node (Domino™, Sametime, Quickplace®) are optional (see Figure 2-9 on page 38). When integrating WebSphere Commerce with WebSphere Portal, the Directory node is required for single sign-on (SSO). The Web server node, WebSphere Commerce Payments node, and Database node components can be installed on the same system as WebSphere Commerce.

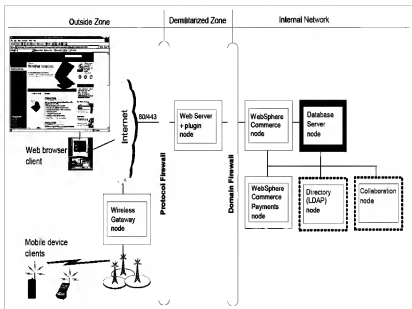


Figure 2-9 WebSphere Commerce runtime architecture and distinguishing features for comparison

WebSphere Commerce mobile device client access

WebSphere Commerce V5.4 can support mobile devices such as a mobile phone, wireless PDA using WAP, i-mode, and HTTP protocols. This was made possible in WebSphere Commerce Suite V5.1 with the introduction of the PvC adapter, supporting database tables, and content-specific JSPs containing a markup language and format that the mobile device can understand. This functionality also exists in WebSphere Commerce V5.4. This type of mobile device access is named mobile commerce direct (or m-commerce direct).

The direct approach uses a device manager that receives a request containing information about a device from a servlet. It determines which adapter would best process the request, and passes the request to the appropriate adapter. Adapters are device-specific components that perform processing functions before passing a request to a controller.

WebSphere Commerce m-commerce direct request flow

The flow of a request from a mobile device to WebSphere Commerce using m-commerce direct (see Figure 2-10) is as follows:

1. To prevent applications from having to handle system functions, such as access control and authentication, requests from any device are first processed by the WebSphere Commerce Web Controller.
2. The adapter creates a session context and a controller request object, and passes the controller request object to the Web Controller.

The controller request object contains a set of properties, formatted by the adapter. It also contains a backward reference to the adapter object and a reference to the session context object created by the adapter.

3. The Web controller executes the request by invoking the corresponding controller command.

All business logic is implemented in the controller command.

4. Based on the view name returned by the controller command, it will return the appropriate JSP file to the requesting device as seen in Figure 2-10.

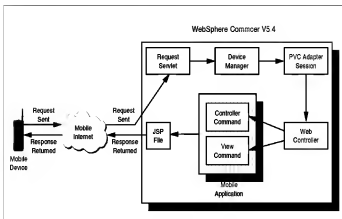


Figure 2-10 m-commerce direct - computing flow

To implement the m-commerce direct approach, you must configure the device manager to recognize the type of device that is assessing your store. This is accomplished by creating a PVC adapter and content JSPs with the appropriate markup language of the targeted mobile device.

Note: The WebSphere Commerce Portal architecture uses the PvC adapter technology and content-specific JSPs for the target mobile device. However, the WebSphere Commerce Portal HTTP flow is different.

WebSphere Commerce components of m-commerce direct

WebSphere Commerce includes the following components to support mobile device clients for m-commerce direct, and also included in the WebSphere Commerce Portal integration solution.

- ▶ **PvC adapter framework**

The PvC adapter provides WebSphere Commerce the ability to detect mobile device types, maintain a session with mobile devices that do not support cookies, and set the root directory for content-specific JSPs containing the supported markup language of the requesting device (HTTP, WAP WML, i-mode cHTML).

In practice, a PvC adapter is created and deployed for different mobile devices protocol types, such as WAP. In addition, a static list of mobile device types need to be supplied to the PvC adapter for device recognition. The PvC adapter compares the value of the User Agent found in the HTTP request of the mobile device with a list of known device types.

The PvC adapter technology is present in both m-commerce direct and WebSphere Commerce Portal.

- ▶ **PvC commands**

The PvC commands are used within the content JSPs for merchants who wish to manage user and device information. The following PvC commands are included with WebSphere Commerce to address the unique requirements of mobile devices outlined in "Unique challenges presented by mobile devices" on page 52:

- PvCRegistration
- PvCRegistrationDevice
- PvCChangeDevice
- PvCBufferUrl
- ReEnterPassword

- ▶ **PvC data beans**

The PvC data beans provide access to buffered parameters and the user device address.

- PvCBufferDataBean

This data bean is required if you want to write a JSP page with parameter buffering.

- UserPVCDeviceDataBean

This data bean allows you to access a user's device address stored in the USERPVCDEV table. You can use this data bean when you need to extract data such as an e-mail address from the USERPVCDEV table.

► **PvC database tables**

The PvC database tables are used for content management, session control and to store information about the user and device. WebSphere Commerce provides the following PvC database tables:

- PVCDEVM DL
- PVCMDLSPEC
- PVCDEVSPEC
- PVCBINDING
- PVCSESSION
- PVCBUFFER

WebSphere Commerce subsystems and components

Figure 2-11 on page 42 depicts the high-level subsystems and components of the WebSphere Commerce Server.

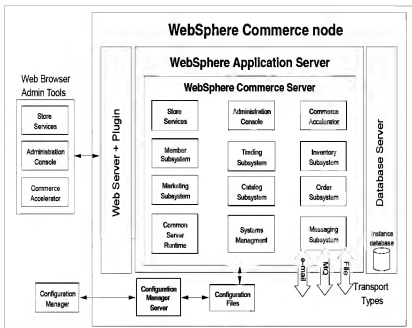


Figure 2-11 WebSphere Commerce Server runtime components

The WebSphere Commerce Server components are categorized as follows (see Figure 2-11):

- WebSphere Commerce subsystems
 - Member subsystem

Note: The Member subsystem is a component of the WebSphere Commerce Server that provides a framework for managing the following participants of the system:

- ▶ Organizational entity (for example, IBM)
- ▶ Organizational unit (for example, IBM Software Group)
- ▶ Member group - group of users
- ▶ Members - users, member groups, organizational entity

The member data is stored within either a WebSphere Commerce instance database or an LDAP directory server database. By default, WebSphere Commerce uses the instance database as its registry. As we will see later, a commerce enabled portal requires that LDAP be used for the user registry, authentication, and single sign-on.

- Catalog subsystem
- Trading subsystem
- Order subsystem
- Marketing subsystem
- Inventory subsystem
- Payment subsystem
- Messaging subsystem
- ▶ WebSphere Commerce Administration tools
 - WebSphere Commerce Configuration Manager
 - WebSphere Commerce Store Services
 - WebSphere Commerce Accelerator
 - WebSphere Commerce Administration Console
- ▶ Common runtime components
 - Common Server runtime
 - Systems management
 - Instance database

WebSphere Commerce HTTP request flow

This section describes the flow of a typical HTTP request from a Web browser client to a WebSphere Commerce Server (see Figure 2-12 on page 44).

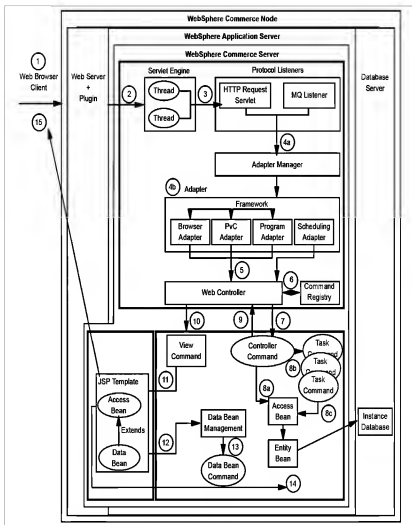


Figure 2-12 WebSphere Commerce flow of an HTTP request

The following information corresponds to the detailed flow of an HTTP request as seen in Figure 2-12:

1. The request is directed to the servlet engine by the WebSphere Application Server plug-in.

2. The request is executed in its own thread. The servlet engine dispatches the request to a protocol listener. The protocol listener can be the HTTP request servlet or the MQ Listener.
3. The protocol listener passes the request to the Adapter Manager.
4. The Adapter Manager determines which adapter is capable of handling the request and then forwards the request to the appropriate adapter. For example, if the request came from an Internet browser, the Adapter Manager forwards the request to the HTTP browser adapter.
5. The adapter passes the request to the Web Controller.
6. The Web Controller determines which command to invoke by querying the command registry.
7. Assuming that the request requires the use of a controller command, the Web Controller invokes the appropriate controller command.
8. Once a controller command begins execution, there are a few possible paths:
 - The controller command can access the database using an access bean and its corresponding entity bean.
 - The controller command can invoke one or more task commands. Then the task commands can access the database, using access beans and their corresponding entity beans (shown in 8c).
9. Upon completion, the controller command returns a view name to the Web Controller.
10. The Web Controller looks up the view name in the VIEWREG table. It invokes the view command implementation that is registered for the device type of the requester.
11. The view command forwards the request to a JSP template.
12. Within the JSP template, a data bean is required to retrieve dynamic information from the database. The data bean manager activates the data bean.
13. The data bean manager invokes a data bean command, if required.
14. The access bean from which the data bean is extended accesses the database using its corresponding entity bean.
15. The JSP writes a response, which is returned to the Web server. The Web server in turn relays that response to the client for the display of the category display page.

2.2.2 WebSphere Portal architecture

By definition, a portal provides access to content, data and services located throughout the enterprise. The key building block for the Portal technology is a portlet. Portlets are an encapsulation of content and functionality. They are reusable components that combine Web-based content, application functionality and access to resources. Portlets are assembled onto portal pages that in turn make up a portal implementation. Portlets are similar to Windows applications in that they present their contents in a window-like display on a portal page. Like a windows application, the portlet window has a title bar that contains controls allowing the users to expand (maximize) and shrink (minimize) the application. Portlets function within the Portal Framework, whereas Windows applications function in the Windows framework. From the portal user's perspective, a portlet is a window on a portal site that provides access to a specific service or resource.

A portal also provides the runtime environment for the portlets that make up the portal implementation. This runtime environment is the portlet container. The portlet container, in the J2EE sense of a container, is responsible for instantiating, invoking, and destroying portlets. The portlet container provides the life cycle infrastructure for the portlets. Portlets rely on their container to provide the necessary infrastructure to support a portal environment. The portal infrastructure provides the core sets of services required by the portlets including:

- ▶ Access to user profile information
- ▶ A framework for portlets to participate in events
- ▶ A framework to communicate with other portlets
- ▶ Access to remote content
- ▶ Access to credentials
- ▶ A framework for storing persistent data.

WebSphere Portal provides these services through an extensible framework for integrating enterprise application, content, people, and processes. WebSphere Portal provides services for single sign-on, security, Web content publishing, search, personalization, collaboration, enterprise application integration, support for mobile devices, and site analysis. Self-service features provided by WebSphere Portal allow users to interact with each other via collaboration, publishing and sharing documents. Users are also provided interfaces for personalizing their own content by managing their own view of the portal, managing their own profiles, and choosing their look and feel.

The WebSphere Portal platform is positioned to enhance the WebSphere family of products providing tooling for aggregating and personalizing Web-based content and making that content available via multiple devices. WebSphere

Portal takes advantage of the strong and flexible platform provided by the WebSphere Applications Server.

WebSphere Portal has its roots in Apache Jetspeed. Jetspeed is an open source implementation of an Enterprise Information Portal, using Java and XML. Jetspeed was created to deliver an Open Source Portal that individuals or companies could use and contribute to in an open source manner.

Building on the Jetspeed implementation WebSphere Portal provides an architecture for building and running portal applications. The overall WebSphere Portal architecture can be seen in Figure 2-13. WebSphere Portal provides services for Authentication and Authorization through its WebSphere Member Services. The core of WebSphere Portal architecture is composed of the presentation services, the portal infrastructure, and the portal services.

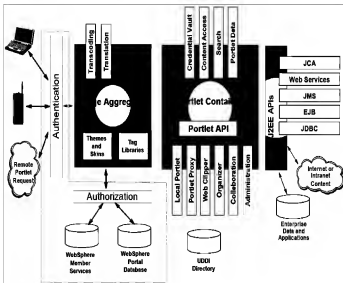


Figure 2-13 WebSphere Portal architecture

Presentation services

WebSphere Portal presentation services provide customized and personalized pages for users through aggregation. Page content is aggregated from a variety of sources via content and applications. The portal presentation framework simplifies the development and maintenance of the portal by defining the page structure independent of the portlet definition. Portlets can be changed without an impact on the overall portal page structure.

Portal Engine

WebSphere Portal provides a pure Java engine whose main responsibility is to aggregate content from different sources and serve the aggregated content to multiple devices. The Portal Engine also provides a framework that allows the presentation layer of the portal to be de-coupled from the portlet implementation details. This allows the portlets to be maintained as discrete components.

Figure 2-14 shows the WebSphere Portal Engine components.

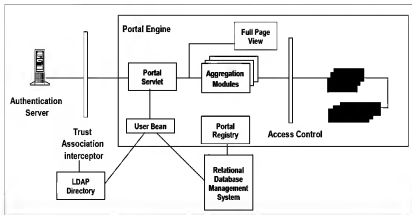


Figure 2-14 WebSphere Portal Engine

The **Authentication Server** is a third-party authentication proxy server that sits in front of the Portal Engine. Access to portlets is controlled by checking access rights during page aggregation, page customization, and other access points.

The **Portal Servlet** is the main component of the Portal Engine. The Portal Servlet handles the requests made to the portal. The portal requests are handled in two phases. The first phase allows portals to send event messages among themselves. In the second phase, the appropriate **Aggregation Modules** for the requesting device renders the overall portal page by collecting information from all the portlets on the page and adding standard decorations such as title bars, edit buttons, etc.

Portlet container

Portal services are components WebSphere Portal uses to extend the portal functionality. Key functionality is provided with WebSphere Portal for personalization, search, content management, site analysis, enterprise application integration collaboration and Web services. Portlets can access these services via their containers.

Portal infrastructure

WebSphere Portal infrastructure is the framework that provides the internal features of the portal. Functionality such as user and group management via self-registration and portal administration are provided by the Portal infrastructure.

User and group management

The WebSphere Portal infrastructure provides facilities to allow user self-management along with enterprise integration with user directories such as LDAP or database structures.

Security services

Since WebSphere Portal runs within the WebSphere Application Server platform, it makes use of the standard Java Security APIs to provide authentication. The WebSphere Portal is configured so that incoming requests pass through an authentication component such as WebSphere Application Server, Tivoli® WebSEAL, or other proxy servers. A user's authorization for a particular resource such as a page or a portlet is handled by the Portal Engine.

User beans are provided to allow programmatic access to the user information for use within portlets.

Page transformation

WebSphere Transcoding Technology is integrated with WebSphere Portal to transform the portal markup produced by WebSphere Portal to markup for additional devices such as mobile phones and PDAs. This technology is not used by the WebSphere Commerce Portal.

Portal services

Portal services are built-in features the WebSphere Portal provides to extend and enhance the full portal solution. These services are provided via the Portlet container as seen in Figure 2-13 on page 47. Among the services are:

- **Personalization**

The IBM WebSphere Personalization functionality enables advanced personalization capabilities. Base customization, such as choosing which portlets are desired on a page, is accomplished by the user via administration functionality. Advanced personalization via rules engines, user preferences, and profiles is accomplished by the provided personalization services.

- **Content Management**

WebSphere Portal provides services to facilitate connections to content management sources. Built-in support is provided for several common content types such as Rich Site Summary (RSS), News Markup Language

(NewsML) and Open Content Syndication (OCS), along with most XML and Web browser markup.

► **Search**

WebSphere Portal offers a simple search service. The Portal Search capability enables search across distributed HTML and text data sources. The search can crawl a Web site and is configured to force it to follow several layers in a site or to extend beyond several links in a site. Furthermore, IBM Extended Search and Enterprise Information Portal can be fully incorporated into the portal environment. These search engines are industrial-strength tools that provide federated searches across numerous data sources.

► **Site Analysis**

You can take advantage of the underlying WebSphere Application Server technology and Site Analyzer to provide information about Web site visitor trends, usage, and content. This detailed information can then be used to improve the overall effectiveness of the site.

► **Collaboration**

Collaboration services are provided by WebSphere Portal through a set of predefined portlets. These portlets allow for team-room function, chat, e-mail, calendaring, and many other collaborative technologies.

► **Web services**

WebSphere Portal provides services for exposing and integrating portlets as remote portlets hosted on another portal platform via Web services technology. The entire process of packaging and responding to a SOAP request is hidden from the developer and the administrator.

WebSphere Portal tooling

WebSphere Portal and the WebSphere Portal Toolkit along with their prerequisite products provide the basic tooling for developing deploying portals and their associated portlets.

WebSphere Portal

WebSphere Portal contains built-in support for portlet deployment, configuration, administration, and communication between portlets.

WebSphere Portal provides the framework for building and deploying portals and the portal components called *portlets*. Portlet content is aggregated by the WebSphere Portal to provide the desired portal implementation.

WebSphere Portal makes use of the WebSphere Application Server technology to provide a portal platform.

WebSphere Portal Toolkit

The WebSphere Portal Toolkit is provided with WebSphere Portal and provides an environment for developing portals using WebSphere Portal. The WebSphere Portal Toolkit is a plug-in for WebSphere Studio Application Developer or WebSphere Studio Site Developer, which adds the portal development environment.

The WebSphere Portal Toolkit provides the ability to quickly create complete MVC-compliant portlet applications. It also provides intuitive editors for working with the deployment descriptors required by your portlet applications. Furthermore, it allows you to dynamically debug your portlet applications.

Where to find more information

To find more detailed information on WebSphere Portal, refer to the following Redbooks:

- ▶ *WebSphere Portal V4.1 Handbook Volume 1*, SG24-6883
- ▶ *WebSphere Portal V4.1 Handbook Volume 2*, SG24-6920
- ▶ *WebSphere Portal V4.1 Handbook Volume 3*, SG24-6921
- ▶ *WebSphere Portal V4.1 Developer's Handbook*, SG24-6897
- ▶ *Mobile Applications with IBM WebSphere Everyplace Access Design and Development*, SG24-6259
- ▶ *Access Integration Pattern using IBM WebSphere Portal Server*, SG24-6267

2.3 Mobile client access

Over the past couple of years, mobile computing has increased dramatically and many customers are in need of solutions for providing mobile client access to commerce sites for mobile commerce. Simply put, mobile commerce, or *m-commerce*, is electronic commerce using mobile devices such as a mobile phone or a wireless PDA.

IBM has developed two solutions for providing mobile device access to WebSphere Commerce Web sites, mobile commerce direct, and commerce enabled portals. In either case, the mobile devices equipped with a micro Web browser will be served content in a markup language the mobile device understands, such as WAP WML or i-mode cHTML. WebSphere Commerce can also communicate to clients using SMS or e-mail. However, in this redbook we limit our mobile device support discussion to mobile clients with a micro Web browser accessing a commerce enabled portal site.

Both WebSphere Commerce and WebSphere Portal include an architecture to support mobile devices. For legacy reasons, WebSphere Commerce Portal uses the PVC adapter technology included with WebSphere Commerce to support mobile devices.

Unique challenges presented by mobile devices

This section describes the unique challenges presented by mobile devices, as well as a functionality overview of the features added to enable WebSphere Commerce for mobile commerce.

► Session control

Session control provides a means for a Web server to keep track of the client system users' transactions, valid URLs, and preferences. On a standard Web browser client, session control is often maintained by using a cookie in cooperation with the browser, such as Microsoft Internet Explorer or Netscape Navigator.

In the mobile computing world, some mobile devices do not support cookies on the client device. In other cases, the session may be maintained by the service provider protocol gateway (for example, WAP).

For this reason, the Web server needs another way to identify such a cookie-less device as a client.

WebSphere Commerce supports a function that enables session control between a Web server and mobile device by using a unique identifier stored in a WebSphere Commerce PVC session table called PVCSESSION. The unique identifier information is sent in the HTTP request header provided by the mobile telecommunication carrier or created automatically by the PVC Adapter.

► Device control

There are several categories of mobile devices such as mobile phones, PDAs, and wireless laptops. Within the categories of mobile devices, there are often unique characteristics and specifications. For example, two WAP phones, whether from the same manufacturer or different manufacturers, may vary from each other in the following features:

- Microbrowser
- Size of screen
- Screen resolution
- Color capability
- Size of memory

WebSphere Commerce provides the ability to detect the mobile device type via the incoming HTTP request from the mobile device, and send back the JSPs with the appropriate content and images for that mobile device. This

requires content JSPs and images with the appropriate markup language and graphics to support the target mobile device.

- Security enhancement

A mobile device is easily lost or stolen because it is generally small. Also, in light of the portability of mobile devices, the chances of your user ID and password being seen in a public space increase. To address these security issues, WebSphere Commerce includes enhanced security functions for logon timeout, command execution restriction, and user registration control.

- URL buffering

Generally speaking, there is a limitation on the length of HTTP requests that can be transmitted from a small device such as a portable cell phone. Lengthy URLs with parameter contents, for example the registration input form for a user profile, may be cut in half due to the small memory in the device. To resolve this problem, WebSphere Commerce provides URL buffering functions. By using the `PVCBufferUrl` function, we can divide the original content (perhaps content originally designed for PC browsers) into several pieces, and send these pieces to the WebSphere Commerce Server separately. The WebSphere Commerce Server temporarily buffers the separate data in the WebSphere Commerce instance database. After all data is sent, the command is executed using the buffered data.

Note: IBM has developed two methods of providing mobile device solutions for WebSphere Commerce Web sites:

- Commerce enabled portals

This redbook includes an example of providing WAP mobile devices access to a commerce enabled portal (see Chapter 10, "Creating a commerce enabled portal for mobile clients" on page 347).

- Mobile commerce direct

This method was introduced in WebSphere Commerce Suite V5.1 and is offered out of the box in WebSphere Commerce V5.4. In this solution, a mobile adapter such as WAP or i-mode is implemented to detect the incoming device type and set the root directory of the JSPs containing the native markup language (HTML, WAP WML, i-mode cHTML) of the requesting mobile device.

This method is not discussed in this redbook. For more information on this technology, refer to the *Mobile Commerce Solutions Guide*, using *WebSphere Commerce Suite V5.1*, SG24-6171 redbook.

WebSphere Commerce Portal mobile client architecture

Both WebSphere Commerce and WebSphere Portal include technologies that can detect the device type of the HTTP request. The WebSphere Commerce Portal mobile client access architecture leverages the PvC adapter technology included with WebSphere Commerce V5.4 (introduced in WebSphere Commerce Suite V5.1).

The PvC adapter architecture performs the following functions:

- Detects the device model type of the HTTP request User Agent
- Sets the JSP directory root for the target device type
- Retrieves the terminal ID used to establish session control for the mobile device

Figure 2-15 depicts the high-level HTTP flow for a mobile client accessing a commerce-enabled store.

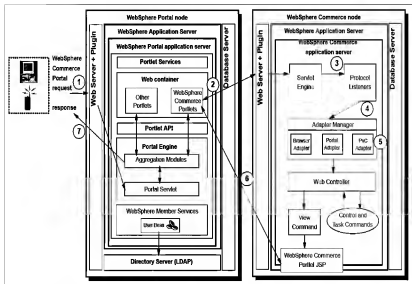


Figure 2-15 WebSphere Commerce Portal mobile client access HTTP flow

The HTTP flow of the request from a mobile client is very similar to that of the Web browser client when accessing the commerce enabled portal. The following

are the key steps in the HTTP flow for emphasizing the mobile access features in the solution:

1. Mobile device client, such as a WAP phone, accesses the commerce enabled portal.
2. Like the standard Web browser client, the commerce portlet (WCSRemoteServletInvoker) interacts with the Web server of the WebSphere Commerce node, which in turn redirects the request to the WebSphere Commerce Servlet Engine.
3. The Protocol Listener identifies that the request is an HTTP request and sends the request to the Adapter Manager. The Adapter Manager determines which adapter is capable of handling the request and then forwards the request to the appropriate adapter. For example, if the request came from a standard Web browser client, the Adapter Manager forwards the request to the HTTP Browser Adapter. If the request came from a commerce portlet, it forwards the request to the Portal Adapter. If the request came from a mobile device, it forwards the request to the PvC Adapter.
4. In the case of a mobile device client, the PvC Adapter is used (for example, WAP Adapter).
5. The PvC adapter is capable of detecting mobile device models based on a static list of mobile device models it is aware of.

The PvC adapter compares the value of the User Agent found in the HTTP header of the request with the static list.

If the requesting mobile device User Agent is found within the PvC known adapters list, the PvC adapter will set the JSP root directory for the device. The JSP root directory contains the commerce portlet JSPs for the desired mobile device such as WAP. In this example, the JSPs include WML markup language which is understood by the WAP device.

6. The commerce portlet JSP containing WML markup is returned to the commerce portlet.
7. The Portlet Aggregation Module aggregates and renders the page for the WAP mobile device and returns the page to the mobile device client.

2.3.1 Where to find more information

Additional information on mobile device client access can be found as follows:

- WebSphere Commerce
 - *Mobile Commerce Solutions Guide, using WebSphere Commerce Suite V5.1*, SG24-6171

- ▶ WebSphere Portal
 - *WebSphere Portal V4.1 Developer's Handbook*, SG24-6897
 - *Mobile Applications with IBM WebSphere Everyplace Access Design and Development*, SG24-6259
- ▶ WebSphere Commerce Portal

Refer to Chapter 10, "Creating a commerce enabled portal for mobile clients" on page 347.

2.4 Authentication and single sign-on

By definition authentication is the process of verifying the identity of a user who is logging onto a computer system. When implementing a commerce enabled portal solution, consideration needs to be given for the authentication methods provided by WebSphere Commerce and WebSphere Portal and how these methods can be implemented for single sign-on.

WebSphere Commerce supports two modes of authentication: WebSphere Commerce instance database or LDAP directory. WebSphere Portal supports three authentication modes: database only, database and LDAP directory, and custom user registry mode. To enable single sign-on for the WebSphere Commerce Portal solution, WebSphere Commerce must be configured to use a common LDAP directory, and WebSphere Portal configured to use the database and LDAP directory mode.

The WebSphere Commerce Portal solution includes a commerce portal login command (`AccessControlUpdateLoginCommand`) as a replacement for the WebSphere Portal login command. The commerce portal login command is only needed to enable commerce personalization features beyond the capability of the standard portal login.

Where to find more information

Additional information on authentication can be found in this chapter as follows:

- ▶ WebSphere Commerce
 - *WebSphere Commerce V5.4 Handbook*, SG24-6567 chapter on IBM SecureWay® Directory (LDAP) integration.
 - 5.3, "Directory Server (LDAP) node implementation" on page 199
- ▶ WebSphere Portal
 - *WebSphere Portal V4.1 Handbook Volume 1*, SG24-6883
 - *WebSphere Portal V4.1, Windows 2000 Installation*, REDP3593

- WebSphere Commerce Portal
 - Chapter 5, “Implement the runtime environment” on page 165
 - *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*

2.5 Access control

Access control refers to management of permissions of a user that has been authenticated on the Web site. This may include limiting the way in which the user or organization interacts with the site.

There are several access control types implemented for WebSphere, WebSphere Commerce, WebSphere Portal, and WebSphere Commerce Portal. The WebSphere Application Server relies on a combination of the Tivoli Access Manager (TAM) and IBM Directory Server (LDAP) for access control and authentication services. Both WebSphere Commerce and WebSphere Portal have their own product-specific implementations for access control. When implementing a commerce enabled portal site, it is possible that the Web site may include all three methods of access control.

WebSphere Portal access control

Access control within IBM WebSphere Portal V4.2.1 is managed from the Portal Administration available from the pull-down found on the WebSphere Portal home page. This allows the administrator to set the access permissions (none, view, edit, manage, delegate) for users and groups for a given object type (portlets, portlet application, portal, places, etc.).

WebSphere Commerce access control

Once the organizations and users that will participate with the WebSphere Commerce site have been defined, the activities performed can be restricted as desired by a set of access control policies. The activities that access control policies operate can range from registration, managing auctions, updating the product catalog, granting approvals on orders, as well as hundreds of other activities required to manage an e-commerce Web site. In a nutshell, the access control policies are the permissions that grant access to various features of your e-commerce Web site.

From a programming perspective, some common functions of access control regarding a WebSphere Commerce site are the access controls for protecting views, commands, and data bean execution.

The following is a summary of the access control methods included in WebSphere Commerce:

- ▶ **User interface**

In addition to the policy editing pages accessible from the Access Management menu of the Administration Console, WebSphere Commerce provides view pages for viewing policies, and their related action groups, access groups, and resource groups. The policy viewing pages are seamlessly integrated into the Administration Console user interface and can be accessed using the buttons added to the existing policy editing pages.

- ▶ **Coarse and fine-grained access control**

WebSphere Commerce V5.4 provides the ability for coarse-grained and fine-grained access control.

WebSphere Commerce coarse-grained access control can be used, for example, to permit buyers to cancel orders by invoking the cancel order function.

WebSphere Commerce provides the ability for fine-grained access control by defining who can invoke what functions against which business object instances (also referred to as resources). In the same example, you are not only able to permit buyers to cancel orders, but limit buyers to invoke the cancel order function only against their own orders, not the orders of other users.

The added power of fine-grained access control combined with coarse-grained access control allows you a greater range of access management and the ability to fine-tune the activities that users are permitted to do on your site.

Both coarse-grained and fine-grained access control can be implemented using XML-based access control policies, which can be modified from the Policy Viewer of the WebSphere Commerce Administration Console.

WebSphere Commerce Portal

The access control permissions for a WebSphere Commerce Portal solution include both WebSphere Portal permissions and WebSphere Commerce access controls. The commerce enabled portal store developed will need to incorporate the access control methods of both WebSphere Commerce and WebSphere Portal.

Where to find more information

Additional information on access control can be found as follows:

- ▶ **WebSphere Commerce**
 - *Access Control Guide, IBM WebSphere Commerce V5.4 product guide*

- ▶ WebSphere Portal
 - *WebSphere Portal V4.1 Handbook Volume 3*, SG24-6921
 - *WebSphere Portal V4.1 Developer's Handbook*, SG24-6897
- ▶ WebSphere Commerce Portal
 - Chapter 5, “Implement the runtime environment” on page 165

2.6 Session management

Session management refers to the management of session information that allows a Web application to maintain state information across multiple user visits to the application. In many Web applications, users dynamically collect data as they navigate the site based on a series of selections of the pages they visit. The navigation path of a Web site by a user may determine what the Web application will display on the next page. For example, if a user clicks the Checkout button on an e-commerce site, the next page must contain the user's shopping selections.

In order to manage the session state, the Web application needs a mechanism to hold the user's state information over a period of time. HTTP is a stateless protocol, meaning the session alone does not recognize or maintain a user's state. HTTP treats each user's request as a discrete independent interaction.

WebSphere Application Server session management

The WebSphere Application Server provides two methods of addressing session management when using the Java servlet specification, cookie, and URL encoding (also known as URL rewriting). Both of these session management methods allow the Web application to maintain all user state information at the host, while passing minimal information back to the client.

WebSphere Commerce session management

WebSphere Commerce is a WebSphere application that is based on the J2EE specification. For this reason, WebSphere Commerce follows the servlet specification for session management. When configuring WebSphere Commerce for session management, the administrator will have the option of selecting session manager and the session type.

From the WebSphere Commerce Configurator, the administrator can select either WebSphere Commerce or WebSphere Application Server as the session manager. The WebSphere Commerce session manager offers better performance, but does not allow extra information to be added to the session and the WebSphere Application Server does.

In addition, the administrator can also select the session type, cookie-based or URL encoding. For security reasons, cookie-based session management uses two types of cookies: non-secure session cookies, which are used to manage the session data, and secure authentication cookies, used to manage authentication data.

On standard Web browser clients, the session is often maintained by using a cookie in cooperation with the browser, such as Microsoft Internet Explorer or Netscape Navigator. In the mobile computing world, some mobile devices do not support cookies on the client device. In other cases, the session may be maintained by the service provider protocol gateway (for example, WAP). For this reason, the Web server needs another way to identify such a cookie-less device as a client.

WebSphere Commerce V5.4 supports a function that enables session control between a Web server and mobile device by using a Pvc adapter and a unique identifier stored in a WebSphere Commerce Pvc session table called PVCSESSION. The unique identifier information is sent in the HTTP request header provided by the mobile telecommunication carrier.

WebSphere Portal session management

The IBM WebSphere Portal product includes a development framework based on the Portlet API V1.1. A portlet is central to portal development. The portlet API includes an abstract Portlet class that extends HttpServlet of the Java Servlet API. All portlets extend the Portlet class indirectly and inherit HttpServlet. For this reason, WebSphere Portal applications share the session management methods provided by the WebSphere Application Server (cookies and URL encoding). In place of HttpRequest, HttpResponse, and HttpSession used for servlets, a portal developer will use the methods PortletRequest, PortletResponse, and PortletSession, which the Portlet class has extended HttpServlet.

WebSphere Commerce Portal session management

The WebSphere Commerce Portal session management does share the characteristics of WebSphere Commerce and WebSphere Portal, but has a unique feature of its own to maintain a session.

The WebSphere Commerce Portal modifies the portal mechanism for maintaining session information. A typical commerce portal page will contain several portlets. For example, a commerce portal page may contain a category display, product display, and shopping cart. The content and data displayed in each portlet may be dependent on HTTP request/response with the WebSphere Commerce Server.

Whenever a WebSphere Portal user uses a commerce enabled portal application, the instance data is associated with the user session. The added

WebSphere Commerce Portal functionality manages a `SessionObject`, which is shared by all commerce portlets of the WebSphere Portal user. This `SessionObject` provides services to maintain session relevant data to issue a WebSphere Commerce HTTP request. The commerce portlet internally sends a WebSphere Commerce HTTP requests using the `HttpClient`. The requests to the WebSphere Commerce server are serialized in the needed order to prevent a problem with the sequence of the request/response of each commerce portlet being served out of order for what is needed for the commerce portal page.

Where to find more information

Additional information on access control can be found as follows:

- ▶ WebSphere Commerce
 - WebSphere Commerce V5.4 online documentation
 - *WebSphere Commerce V5.4 Handbook*, SG24-6567
- ▶ WebSphere Portal
 - *WebSphere Portal V4.1 Developer's Handbook*, SG24-6897

2.7 Presentation services and page rendering

With traditional Web applications, the client is the Web browser and interaction with the server is performed using HTTP requests and responses. In the case of a portal, the portal becomes the client, since it is performing the aggregation of the content from the portlets and rendering of the page sent to the Web browser client.

Traditional WebSphere Commerce Web sites and WebSphere Commerce Portal sites serve Web browser clients HTML-based display pages using JavaServer Page (JSP) technology. Web browser clients can run on Windows, UNIX, Linux, or MAC operating systems. The most common Web browsers used are Microsoft Internet Explorer and Netscape Navigator/Communicator. Each JSP served to the Web browser client contains HTML tags such as `<header>`, `<html>`, and `<body>` used by the Web browser to render the page.

In the case of a WebSphere Portal application, the portal server assembles the contents of the individual portlets and renders the page, which is sent to the Web browser client. For this reason, the portlet JSPs do not have formatting tags found in a standard JSP, such as `<header>`. The portlet JSPs can contain HTML or markup languages for mobile devices such as WML.

Where to find more information

Additional information on presentation services and page rendering can be found as follows:

- ▶ WebSphere Portal
 - WebSphere Portal V4.1 Developer's Handbook*, SG24-6897
- ▶ WebSphere Commerce Portal
 - *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*, in the appendix on converting existing JSPs
 - 7.3, "Customizing the ITSO B2B CEP store" on page 291



Patterns for e-business and commerce enabled portals

The job of an IT architect is to evaluate business problems and to develop a solution. The architect begins by gathering input on the problem, the desired solution, and any special considerations or requirements that need to be factored in. The architect takes this input and designs a solution that includes one or more applications that provide the necessary functions and services.

It is to our advantage to capture the experience of these IT architects in such a way that future engagements are made simpler. Taking this experience and crafting a repository that provides a way for architects to use this experience to build future solutions using proven scenarios saves time and money for everyone and helps ensure a solid solution that will stand the test of time. The IBM Patterns for e-business effort does just this. Its purpose is to capture e-business approaches that have been tested and proven. The information captured is thought to fit the majority of situations.

This chapter provides an introduction to the IBM Patterns for e-business and describes the Electronic Commerce composite pattern.

Note: The IBM Commerce Enhancement Pack - April 2003 Edition provides commerce enabled portal integration code for WebSphere Commerce V5.4.0.5 and WebSphere Portal V4.2.1 Enable.

3.1 Patterns for e-business

The IBM Patterns for e-business are a collective set of proven architectures that have been compiled from successful Internet-based engagements. This repository of assets can be used by companies to facilitate the development of Web-based applications. They help an organization understand and analyze complex business problems and break them down into smaller, more manageable functions that can then be implemented using low-level design patterns.

This section describes the fundamentals of Patterns for e-business, how they enable architects to build successful e-business solutions, and where to find more detailed information.

3.1.1 Introduction to Patterns for e-business

As companies compete in the e-business marketplace, they find that they must re-evaluate their business processes and applications so that their technology is not limited by time, space, organizational boundaries, or territorial borders. They must consider the time it takes to implement the solution, as well as the resources (people, money, and time) they have at their disposal to successfully execute the solution. These challenges, coupled with the integration issues of existing legacy systems and the pressure to deliver consistent high-quality service, present a significant undertaking when developing an e-business solution.

In an effort to alleviate the tasks involved in defining an e-business solution, IBM has built a repository of "patterns" to simplify the effort. In simple terms, a pattern can be defined as a model or plan used as a guide in making things. As such, patterns serve to facilitate the development and production of things. Patterns codify the repeatable experience and knowledge of people who have performed similar tasks before. Patterns not only document solutions to common problems, but also point out pitfalls that should be avoided. IBM's Patterns for e-business consists of documented architectural best practices. They define a comprehensive framework of guidelines and techniques that are used in creating architectures for customer engagements. The Patterns for e-business bridge the business and IT gap by defining architectural patterns at various levels, from Business patterns to Application patterns to Runtime patterns, enabling easy navigation from one level to the next. Each of the patterns (Business, Integration, Application, and Runtime) help companies understand the true scope of their development project and provide the necessary tools to facilitate the application development process, thereby allowing companies to shorten time to market, reduce risk, and more important, realize a more significant return on investment.

The major categories of Patterns for e-business are:

- ▶ Business patterns
- ▶ Integration patterns
- ▶ Composite patterns
- ▶ Application patterns
- ▶ Runtime patterns and matching Product mappings

When a company takes advantage of these documented assets, they are able to reduce the time and risk involved in completing a project.

For example, a line-of-business (LOB) executive who understands the business aspects and requirements of a solution can use Business patterns to develop a high-level structure for a solution. Business patterns represent common business problems. A LOB executive can match their requirements (IT and business drivers) to Business patterns that have already been documented. The patterns provide tangible solutions to the most frequently encountered business challenges by identifying common interactions among users, business, and data.

Senior technical executives can utilize Application patterns to make critical decisions related to the structure and architecture of the proposed solution. Application patterns help refine Business patterns so that they can be implemented as computer-based solutions. Technical executives can use these patterns to identify and describe the high-level logical components that are needed to implement the key functions identified in a Business pattern. Each Application pattern would describe the structure (tiers of the application), placement of the data, and the integration (loosely or tightly coupled) of the systems involved.

Finally, solution architects and systems designers can develop a technical architecture by using Runtime patterns to realize the Application patterns. Runtime patterns describe the logical architecture that is required to implement an Application pattern. Solution architects can match Runtime patterns to existing environment and business needs. The Runtime pattern they implement establishes the components needed to support the chosen Application pattern. It defines the logical middleware nodes, their roles, and the interfaces among these nodes in order to meet business requirements. The Runtime pattern documents what must be in place to complete the application, but does not specify product brands. Determination of actual products is made in the Product mapping phase of the patterns.

In summary, Patterns for e-business capture e-business approaches that have been tested and proven. By making these approaches available and classifying them into useful categories, LOB executives, planners, architects, and developers can further refine them into useful, tangible guidelines. The Patterns and their associated guidelines allow the individual to start with a problem and a

vision, find a conceptual pattern that fits this vision, define the necessary functional pieces that the application will need to succeed, and then actually build the application. Furthermore, the Patterns for e-business provide common terminology from a project's onset and ensure that the application supports business objectives, significantly reducing cost and risk.

3.1.2 Patterns for e-business layered asset model

The Patterns for e-business approach enables architects to implement successful e-business solutions through the re-use of components and solution elements from proven successful experiences. The Pattern approach is based on a set of layered assets that can be exploited by any existing development methodology. These layered assets are structured in a way that each level of detail builds on the last. These assets include:

- ▶ Business patterns that identify the interaction between users, businesses, and data.
- ▶ Integration patterns that tie multiple Business patterns together when a solution cannot be provided based on a single Business pattern.
- ▶ Composite patterns that represent commonly occurring combinations of Business patterns and Integration patterns.
- ▶ Application patterns that provide a conceptual layout describing how the application components and data within a Business pattern or Integration pattern interact.
- ▶ Runtime patterns that define the logical middleware structure supporting an Application pattern. Runtime patterns depict the major middleware nodes, their roles, and the interfaces between these nodes.
- ▶ Product mappings that identify proven and tested software implementations for each Runtime pattern.
- ▶ Best-practice guidelines for design, development, deployment, and management of e-business applications.

These assets and their relation to each other are shown in Figure 3-1 on page 67.

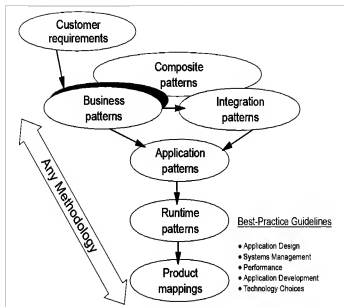


Figure 3-1 The Patterns for e-business layered asset model

3.1.3 Patterns for e-business Web site

The Patterns Web site provides an easy way of navigating top down through the layered Patterns' assets in order to determine the preferred reusable assets for an engagement.

The Patterns for e-business Web site is found at:

<http://www.ibm.com/developerWorks/patterns/>

Additional information about Patterns for e-business is found at:

<http://www.redbooks.com>

3.2 Electronic Commerce composite pattern

Organizations strive to achieve the best combination of deep customer knowledge and mindshare, product leadership, and transactional efficiency as best suits their business goals. In order to obtain these goals, organizations

leverage IT systems to provide various types of information to specific audiences.

The Electronic Commerce composite pattern described in this chapter includes elements of the Sell-Side Hub composite pattern, Portal composite pattern and the Access Integration pattern.

For more information on these patterns, refer to the following redbooks:

- ▶ *B2B e-commerce With WebSphere Commerce Business Edition V5.4, Patterns for e-business Series*, SG24-6194
- ▶ *A Portal Composite Pattern Using WebSphere Portal V4.1.2*, SG24-6869
- ▶ *Access Integration Pattern using IBM WebSphere Portal Server*, SG24-6267

3.2.1 Defining the Electronic Commerce composite pattern

A Composite pattern can be comprised of various combinations of Business, Integration, Application and Runtime patterns, as defined on the "Patterns for e-business" Web site at:

<http://www.ibm.com/developerworks/patterns>

Composite patterns combine Business patterns and Integration patterns to create complex, advanced e-business applications."

The Business and Integration patterns that can be included in the Electronic Commerce composite pattern are as follows:

- ▶ Access Integration
- ▶ Self-Service
- ▶ Collaboration
- ▶ Information Aggregation
- ▶ Extended Enterprise
- ▶ Application Integration

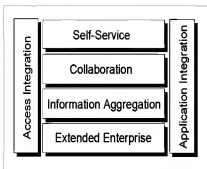


Figure 3-2 Electronic Commerce composite pattern

Depending on the type of commerce enabled portal solution being deployed, different combinations are implemented based on the required functionality. Some of these Business and Integration patterns are required and some are optional when applied to a Electronic Commerce composite pattern.

Access Integration pattern

The Access Integration pattern is commonly observed in e-business solutions that provide a seamless and consistent user experience that combines access to multiple applications, databases, and services. It is used as a front-end integration pattern. The Access Integration pattern does not stand alone in a solution, but is typically used to combine Business patterns to create custom designs and Composite patterns are used to solve complex business problems.

Business and IT drivers

Businesses developing a solution needing the following characteristics should consider using the Access Integration pattern:

- ▶ The end users and customers need to directly interact with business processes.
- ▶ The business activity has a need to aggregate, organize, and present information from various sources within and outside of the organization.
- ▶ The business process must be reachable in a common, consistent, and simplified manner through multiple delivery channels.

The Access Integration pattern typically consists of users who:

- ▶ May be within the enterprise, in business partner organizations, or in any other location across the globe

- ▶ Will typically access the solution using a Web browser or a browser-based Internet appliance. The enterprise has very little or no control over how this device is set up or configured.
- ▶ Can access the solution from any location across the Internet

Access Integration services can include one or more of the following services:

- ▶ Device support
- ▶ Presentation
- ▶ Personalization
- ▶ Security and Administration

Access Integration contains many of the characteristics that describe a portal implementation. It fits well into the Electronic Commerce composite pattern because it includes aggregation and management of information, access to information by various user and group types, and the business "rules" have been clearly defined that determine which user types can access certain types of data.

Note: For more information on the Access Integration pattern and its services, refer to the *Access Integration Pattern using IBM WebSphere Portal Server, SG24-6267* redbook.

Self-Service business pattern

Often an organization not only wants to disseminate information internally but also wants to make this information available to external users and partners. The Self-Service business pattern is focused on allowing the end user access to information from various data sources using a mechanism that allows the user to access just the specific information that applies.

Collaboration business pattern

The Collaboration business pattern enables interaction and collaboration between users. This pattern can be observed in solutions that support small or extended teams who need to work together in order to achieve a joint goal.

Business and IT driver

Businesses developing a solution needing the following characteristics should consider using the Collaboration business pattern:

- ▶ The end users and customers need to directly interact with business processes.
- ▶ The business activity demands and fosters collaboration and the sharing of information among its participants.

The Collaboration business pattern occurs in e-business solutions that involve one-way or two-way interactions between users of the solution. These interactions can take many forms, including:

- ▶ Asynchronous collaboration, in which a user addresses a message to another user or group of users on the network. This message is then sent to a collector (or container) where the intended recipient of the message picks it up. This type of communication is typically seen in traditional e-mail systems that are based on SMTP (Simple Mail Transfer Protocol) and POP (Post Office Protocol) as well as in collaboration Web sites.
- ▶ Interactive collaboration, in which a user collaborates with one or more users by sharing information synchronously. This type of communication is typically implemented through services such as interactive chat rooms, bulletin boards and instant messaging services.
- ▶ Broadcast and multi-casting, in which a user sends a message or a sequence of messages to multiple recipients. This includes support for broadcasting rich media such as audio and video, streaming media, and so on.

The types of collaboration identified above are often combined with a Workflow engine that provides the ability to set up and sequence atomic activities to support more complex processes that might involve multiple users from different workgroups, departments, and organizations. Collaboration is a core feature of a portal implementation.

Information Aggregation business pattern

Information Aggregation is a core feature and purpose of a portal implementation. It brings together data from disparate data sources into a single view, targeted to specific users and groups. The Information Aggregation business pattern (and associated Application and Runtime patterns) contains these characteristics so it fits well with a portal implementation.

Extended Enterprise business pattern

The focus of the Electronic Commerce composite pattern is to implement a portal within a commerce business or single enterprise. It does not directly address how two separate enterprises will interact. In this chapter, our analysis revealed that treating external enterprises as just additional "data sources" seems clearer than talking about enterprise-to-enterprise interaction (for example, covered in the Application Integration patterns domain). However, this is open to interpretation. Portals are about integrating data and processes, so this pattern only makes sense when bringing together the data sources and systems in two enterprises. Thus, this implies a more complex re-architecture of two systems. It is just as effective and less complex to simply treat other external systems as just data sources, the same as local databases or applications. If these external

systems support common communication methods, then this makes the integration that much easier.

Application Integration pattern

A portal will have impacts on the business processes and will also pull and aggregate data from existing applications. These characteristics are core to the Application Integration pattern so this pattern is a good fit for the Electronic Commerce composite pattern.

Portal/Commerce characteristics

Figure 3-3 on page 73 was used as part of the process of identifying the Business, Integration and Application patterns that could be combined into an Electronic Commerce composite pattern.

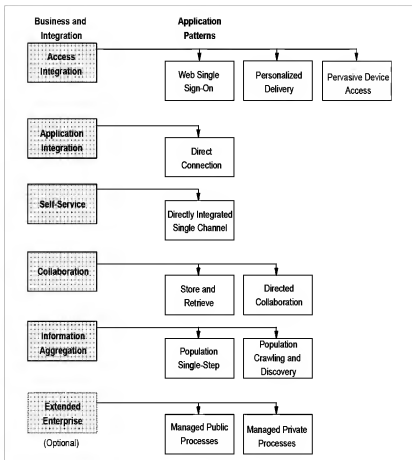


Figure 3-3 Patterns Hierarchy of the Electronic Commerce composite pattern

3.2.2 Identified patterns

The Business and Integration patterns that are identified to be the building blocks to the Electronic Commerce composite pattern are as follows:

- ▶ Access Integration pattern
- ▶ Application Integration pattern
- ▶ Self-Service business pattern
- ▶ Collaboration business pattern

- ▶ Extended Enterprise pattern (optional, but typical for enterprise customer)
- ▶ Information Aggregation pattern
- ▶ Application Integration pattern

The Electronic Commerce Portal composite pattern is a variation of the Electronic Commerce composite pattern, which includes patterns common to the Portal composite pattern. Throughout the redbook, we will simply refer to this as the Electronic Commerce composite pattern.

Please note that based on one's specific requirements, the building blocks of the Business and Integration patterns for the Electronic Commerce composite pattern may vary.

The Electronic Commerce composite pattern for this redbook is illustrated in Figure 3-4.

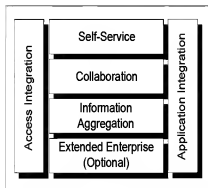


Figure 3-4 Electronic Commerce composite pattern showing the mandatory patterns

3.2.3 Summary

The Electronic Commerce composite pattern is a "best mix" of patterns, technologies, and products. It allows for an understanding of the business and IT drivers that help an organization answer these questions:

- ▶ Do I need a commerce enabled portal?
- ▶ What can I achieve with a commerce enabled portal?

Once an organization has determined that it needs to aggregate information, leverage commerce services, target that information to specific users, analyze the usage of information, and collect and manage information, they can use a commerce enabled portal to handle these requirements. Consequently, using the Electronic Commerce composite pattern will eventually lead to a choice of

Application patterns and the subsequent combined Runtime pattern. This, in turn, will drive the creation of a commerce enabled portal architecture. Some specific benefits include:

- ▶ A single aggregated view of content targeted to specific user types
- ▶ Ability to analyze usage patterns to make marketing efforts more efficient
- ▶ Ability to tailor the user interface to specific groups enabling a focus on cultural, language, and nationality-based differences
- ▶ Single sign-on, allowing the user to save time and have access to information while lessening the requirements for direct interaction with the organization, which saves money

In summary, the Electronic Commerce composite pattern includes characteristics from several Business and Integration patterns that are typically part of a portal implementation. However, when designing your solution, re-evaluate the chosen patterns to assure that they contain the characteristics that are important for the portal solution you are creating. Remember that it is ultimately based on the business drivers and choosing a pattern and subsequent architecture that supports those drivers.

3.3 Selecting the Application pattern

This section describes the Application patterns that apply to the Electronic Commerce composite pattern and includes additional details about how these patterns apply to the ITSO working example scenario in Part 2 of the redbook. Please note that you may need to pick a different Application pattern based on your business requirements.

3.3.1 Access Integration::Web Single Sign-On application pattern

The Web Single Sign-On application pattern (as part of the Access Integration pattern) provides a framework for seamless application access through unified authentication services. Figure 3-5 on page 76 shows an example of this pattern.

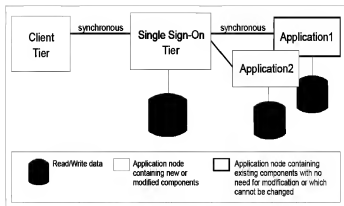


Figure 3-5 Access Integration: Web Single Sign-On pattern

Business and IT drivers

The business and IT drivers are as follows:

- Provide single sign-on across multiple applications.
- Reduce total cost of ownership (TCO).
- Reduce user administration cost.

The primary business driver for choosing this Application pattern is to provide seamless access to multiple applications with a single sign-on while continuing to protect the security of enterprise information and applications.

Simplification and increased efficiency of user profile management is the main IT driver for single sign-on.

Benefits

The benefits are as follows:

- Users can access their application portfolio easily and securely.
- User profile information is centralized in a common directory, simplifying profile management and reducing costs.
- Application development cost is reduced by providing a standard security solution.

Limitations

Many existing applications are not capable of accepting a standard set of user credentials as a substitute for local authentication. Integration with such systems can be difficult or even impossible.

Tip: Single sign-on functionality requires more than just making sure that the applications that already exist in an enterprise support a central authentication capability. The existing processes must be changed to accommodate this new method of validating a user's access capability. An analysis of the existing profiling mechanisms and overall security policies in an organization is the starting point for this type of effort.

Electronic Commerce composite pattern

A fundamental characteristic of a commerce enabled portal implementation is that of commerce-related information aggregation. In order to enhance this experience for the user, a single sign-on (SSO) solution makes sense. This allows the user to more quickly access the information and commerce services and avoid worrying about which application they are accessing. It also allows for easier maintainability by the organization sponsoring the portal. The Electronic Commerce composite pattern accommodates an SSO scenario by means of a separation of the directory and security node and the session level security being leveraged in the application server node.

Working example scenario

The working example scenario for this redbook leverages single sign-on for access to WebSphere Portal Server and WebSphere Commerce separately. This means that they both leverage IBM SecureWay Directory for authentication.

3.3.2 Access Integration::Pervasive Device Access application pattern

The Access Integration pattern is used to provide consistent access to various applications using multiple device types. In order to provide pervasive device access to an existing Business pattern, we therefore need to use an Access Integration application pattern, shown in Figure 3-6 on page 78. The Pervasive Device Access application pattern brings a new tier into the architecture. This tier is responsible for the pervasive extensions to the original application. The function of this tier is to convert the HTML/WML issued by the application presentation logic into a format appropriate for the pervasive device. In this way, the Pervasive Device Access application pattern provides a structure for extending the reach of individual applications from browsers and fat clients to pervasive devices such as PDAs and mobile phones.

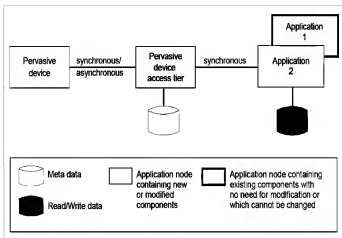


Figure 3-6 Access Integration::Pervasive Device Access

Business and IT drivers

The business and IT drivers are as follows:

- ▶ Provide universal access to information and services
- ▶ Time to market
- ▶ Reduce total cost of ownership (TCO)

Striving to provide universal access to information and applications is often the primary business driver for choosing this Application pattern.

The primary IT driver for choosing this Application pattern is to quickly extend the reach of applications to new device types without having to modify every individual application to enable its use by additional device types.

Electronic Commerce composite pattern

The Electronic Commerce composite pattern supports the use of pervasive device access. In fact, any type of WAP WML device access is supported through the use of templates in the pervasive access device tier. At this tier, the session data containing the type of device is known and the properly formatted content can be delivered. This formatted content can be transcoded in content management, commerce or data source nodes, or it can be transcoded dynamically when requested by a specific type of client. This will depend on the frequency of updates to the data.

Working example scenario

The working example scenario for this redbook demonstrates access for WAP WML-based pervasive devices. We use the OpenWave WML emulator to simulate the WAP mobile device for development and test purposes found at:

<http://www.openwave.com>

Additional templates can be added to provide support for other end-user display formats as needed. Thus, existing commerce JavaServer Pages can be "reused" for different device types.

3.3.3 Access Integration::Personalized Delivery application pattern

The Personalized Delivery application pattern provides a framework for giving access to applications and information tailored to the interests and roles of a specific user or group. This Application pattern extends basic user management by collecting rich profile data that can be kept current up to the user's current session. Data collected can be related to application, business, personal, interaction, or access device-specific preferences. An example of the Personalized Delivery application pattern is shown in Figure 3-7.

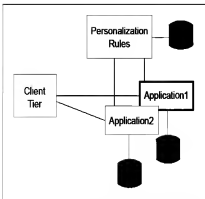


Figure 3-7 Access Integration::Personalized Delivery

Business and IT drivers

The primary business drivers for choosing this Application pattern is to increase usability and improve the efficiency of Web applications by tailoring their presentation to the user's role, interests, habits and/or preferences.

Benefits

The benefits are as follows:

- ▶ Users' interaction with the site is benefited because of increased perception of control and efficiency.
- ▶ Fine-grained control of users' access to applications is enabled according to role and preferences by the enterprise.
- ▶ Improved user effectiveness is enabled by adapting the complexity and detail of content to a user's skill level.

Limitations

The Personalized Delivery application pattern can be very complex and expensive to fully implement.

Electronic Commerce composite pattern

This Application pattern supports the separation of the business logic, business rules, and presentation. Each one of these has part of the responsibility for providing the personalized experience to the user of the portal. The application server handles business logic that implements the business rules meta-data contained in the personalization server node. Once presentation of the personalized data is required, the presentation server node will access the correctly formatted and/or aggregated data for display to the portal user.

Working example scenario

The working example scenario provides the end user with personalized delivery through the use of user preferences and prescriptive personalization (rules and roles based).

3.3.4 Self-Service::Directly Integrated Single Channel application pattern

The Directly Integrated Single Channel application pattern (from the Self-Service business pattern) provides a structure for applications that need one or more point-to-point connections with back-end applications but only need to focus on one delivery channel. This Application pattern shown in Figure 3-8 on page 81 can also be used to implement any one of the delivery channels.

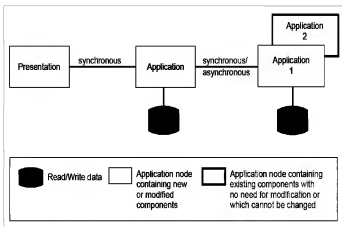


Figure 3-8 Self-Service: Directly Integrated Single Channel

Business and IT drivers

The business and IT drivers are as follows:

- ▶ Improve the organizational efficiency
- ▶ Reduce the latency of business events
- ▶ Leverage existing skills
- ▶ Leverage legacy investment
- ▶ Back-end application integration

The primary business driver for choosing this Application pattern is to reduce the latency of business events by providing real-time access to back-end applications and data from Web applications.

The IT driver for choosing this Application pattern is to leverage legacy investments and existing skills.

Electronic Commerce composite pattern

The Electronic Commerce composite pattern is only lightly involved in the direct connection between the portal user and a commerce application (for example, Lotus® Sametime or WebSphere Commerce-based application). Once the portal user is authenticated via the directory and security services node and the session level security in the application server node, the WebSphere Portal Portlet API will pass authentication credential information to the commerce application or data source. Once complete, the user will now have a direct connection to that application and the portal system will not generally broker the communication.

Working example scenario

The working example scenario focuses on a single commerce Web channel and single WML formatted channel (using a WML emulator) for the purposes of applying the Electronic Commerce composite pattern. It is common for a portal implementation to handle multiple channels simultaneously. Use this pattern to look at the implementation of the portal system for these two channels and re-apply for additional channels. Consider both the Web and WAP WML channels a particular "instance" of the multi-channel pattern. For example, for multiple channels that include a Web browser, wireless connected PDA, and mobile phone, you would have three "instances" of this pattern.

3.3.5 Collaboration: Store and Retrieve application pattern

The Store and Retrieve application pattern (as part of the Collaboration business pattern) allows users to collaborate with others on the network interactively. Unlike the Point-to-Point application pattern, this pattern does not require both partners to be online at the same time. It also does not require the client to know the physical or direct address of other users of the solution. Figure 3-9 shows an example of this Application pattern.

A common implementation of this pattern is the role-based commerce administrative tasks. It allows two or more users to interact on a single component (for example, order data, contract, etc.) via the WebSphere Commerce mechanism.

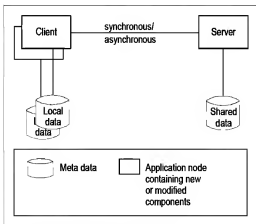


Figure 3-9 Collaboration::Store and Retrieve

Business and IT drivers

The business and IT drivers are as follows:

- ▶ Time to market
- ▶ Improve the organizational efficiency
- ▶ Reduce the latency of business events
- ▶ Easy to adapt during mergers and acquisitions
- ▶ Require deferred collaboration
- ▶ Many users
- ▶ Leverage existing skills
- ▶ Network addressing independence
- ▶ Managed service
- ▶ Maintainability

Guidelines for use

This Application pattern should be used when:

- ▶ The physical or direct addresses of other clients on the network are not known.
- ▶ The pattern can support both synchronous and asynchronous communication. This provides ability to support a wide range of solutions from bulletin boards and workrooms to interactive chat rooms.
- ▶ A server can be set up that will allow multiple clients to log in and share information with other users by posting messages on (or sending e-mail/SMS to) the server for later retrieval.

Benefits

The benefits are as follows:

- ▶ This Application pattern is simple to implement.
- ▶ Since this Application pattern does not require that a client know the direct address of the destination, it is ideal for solutions where the network addresses are not published or where these addresses change frequently.
- ▶ Most of the functions of this pattern can be implemented using commercially available collaboration solutions.
- ▶ This pattern requires very minimal custom code and is cost effective to maintain.

Limitations

The limitations are as follows:

- ▶ This pattern calls for the implementation of server software and associated hardware to support new users. This means that this will add to the overall complexity of the solution
- ▶ The nature and type of collaboration supported by this pattern are simplistic. For more complex communications, later Application patterns are more appropriate.

Electronic Commerce composite pattern

The Electronic Commerce composite pattern supports this through the use of the commerce and collaboration nodes. Commerce can provide asynchronous collaboration on "assets" or documents, and the collaboration can be in the form of threaded discussion forums or teamrooms where information is shared in a common "space".

Working example scenario

In the working example scenario, WebSphere Commerce is used to collect assets from different types of users. These user types are:

1. *Business Customer*, who can send orders to be *approved*
2. *Business Reseller Customer*, who can approve orders that have been *sent*
3. *Site Administrator*, who can notify the customer about approved orders

The "Server" node is represented by WebSphere Commerce, while the "Client" nodes are represented by end users who are accessing the portal. Using WebSphere Commerce, "disconnected" or "asynchronous" collaboration is enabled.

3.3.6 Collaboration::Directed Collaboration application pattern

The Directed Collaboration application pattern allows users to collaborate with others on the network interactively. This Application pattern, shown in Figure 3-10 on page 85, requires the two users who need to interact to be online simultaneously. It also requires users to register with a server. In this pattern, all of the users are peers and there are no client/server or master/slave relationships between the tiers in the pattern.

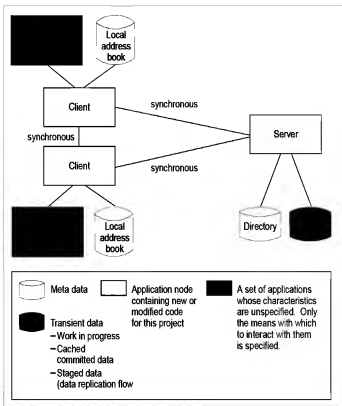


Figure 3-10 Collaboration::Directed Collaboration

Business and IT drivers

The business and IT drivers are as follows:

- ▶ Time to market
- ▶ Improve organizational efficiency
- ▶ Reduce the latency of business events
- ▶ Easy to adapt during mergers and acquisitions
- ▶ Require instantaneous collaboration
- ▶ Many users
- ▶ Leverage existing skills
- ▶ Network addressing independence
- ▶ Managed service
- ▶ Maintainability

- ▶ Complex data types
- ▶ Significant network bandwidth

This approach can be used to quickly establish collaboration between users of a solution without having to go through the process of developing a lot of custom code. It allows users to simultaneously and interactively modify shared applications and data.

This pattern requires all the users to register with the server. The user's profile, preferences, and security privileges are stored on a server directory. This means that the client does not need to know the physical or direct address of other clients. It also allows us to implement different security levels, and implement more complex collaboration styles that include sharing applications and complex data types.

This is the ideal Application pattern to choose if the current focus is to establish synchronous sophisticated collaboration functions within a solution. This solution is also applicable when the clients have permanent and preferably high-speed network connections. The solution is also cost-effective to develop because many of these functions are available in off-the-shelf products.

This pattern is not a good fit for solutions where there are limitations on the processing power of the clients.

Electronic Commerce composite pattern

The collaboration from the Electronic Commerce composite pattern is applied to this type of collaboration. It is generally in the form of instant messaging because communication is essentially a brokered real-time interaction.

Working example scenario

The working example scenario will leverage the Lotus Sametime instant messaging product and the collaborative portlets to provide real-time, interactive communication and collaboration. The single LDAP repository will be used to authenticate a portal user who wants access to the Sametime server. This is an implementation best practice, but in large organizations this can be very complex to implement. The consolidation of security mechanisms and policies can be lengthy from a time, people, and money perspective.

3.3.7 Information Aggregation::Population Single-Step application pattern

The Population Single-Step application pattern structures the population of a data store with data that requires minimal transformation and restructuring. The Population Single-Step application pattern is a simplistic solution design and is

not documented to the Runtime pattern level for the Electronic Commerce composite pattern. An example of the Population Single-Step application pattern is shown in Figure 3-11.

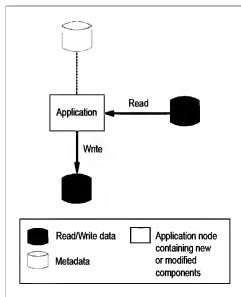


Figure 3-11 Information Aggregation::Population Single-Step

Business and IT drivers

- Improve organizational efficiency
- Reduce the latency of business events
- Distill meaningful information from a vast amount of structure data
- Minimize Total Cost of Ownership (TCO)
- Promote consistency of Operational Data
- Be maintainable

The primary business driver for choosing Population Single-Step is to copy data from the source data store to a target data store with minimal transformation. The main reason for creating a copy of the data is to avoid manipulating the primary source of a company's operational data often maintained by operational systems.

Electronic Commerce composite pattern

The Electronic Commerce composite pattern supports this no-transformation population through a centralized database server. However, in many

installations, data will be transformed before reaching its final destination (for example, database or file system for serving to the Web).

Technical scenario

The technical scenario uses a single-step (no transformation) database population approach for most of the functionality. The data is managed in the content management system (for example, IBM WebSphere Content Publisher) and then "published" in final form to the "CMS Staging DB".

In WebSphere Commerce, catalog data can be loaded to the WebSphere Commerce instance database using the mass loader or by using the WebSphere Commerce Accelerator Web browser based console. The Population Single-Step application pattern applies to scenarios where the WebSphere Commerce catalog data is loaded without the need to be transformed.

For example, a developer may wish to create and load a sample data set for development testing purposes. In addition, when a customer product catalog is relatively small, an XML load file can be created manually and deployed as part of the store archive publishing during the initial deployment.

The Population Multi-Step application pattern is a variation of the Population Single-Step application pattern for scenarios in which content does need to be transformed prior to load. When larger numbers of WebSphere Commerce catalog data need to be managed and content is aggregated from multiple systems, it is common for the data to be transformed using the WebSphere Commerce Loader package (convert, transform, ID resolve, load) and then loaded.

3.3.8 Information Aggregation::Population Crawling and Discovery application pattern

The Population-Crawl and Discovery application pattern provides a structure for applications that retrieve and parse documents and create an index of relevant documents that match specified selection criteria. This design is actually a specific instance of the Population-Single Step application pattern described above. In practice, this design may also extend the Population Multi-Step application pattern, when transformation of data is required. In either case, the crawling and discovery mechanisms of this design aggregate a set of unstructured data. The Population-Crawl and Discovery application pattern shown in Figure 3-12 is a new solution design and is not documented to the Runtime pattern level. This pattern is also useful for solutions where there is a need to discovery content expertise within the organization.

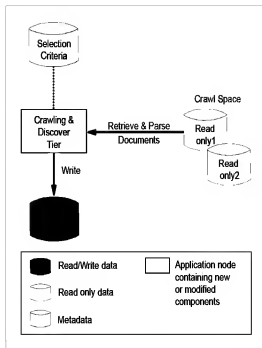


Figure 3-12 Information Aggregation::Population-Crawling and Discovery

Business and IT drivers

- Improve organizational efficiency
- Reduce the latency of business events
- Provide easier access to vast amounts of unstructured data through indexing and categorization
- Reduce information overload
- Identify the experts for collaboration to improve decision cycle times
- Reduce knowledge loss from personnel turnover
- Help new employees to reduce the "learning curve"
- Minimize Total Cost of Ownership
- Promote consistency of Operational Data
- Be maintainable

The primary business driver for choosing Population-Crawl and Discovery is to select relevant documents from a vast set of documents based on specified selection criteria. The objective is to provide quick access to useful information instead of bombarding the user with too much information.

Search engines that crawl the World Wide Web implement this Application pattern. It is best suited for selecting useful information from a huge collection of unstructured text data. A variation of this Application pattern can be used for working with other forms of unstructured data such as images, audio, and video files.

Electronic Commerce composite pattern

In any portal implementation, the ability to locate data and information as it is updated in the system is vital. The whole value proposition depends, in part, on a portal user's ability to locate the information they need. The Electronic Commerce composite pattern supports this through the Search and Indexing node. This represents both the ability to "free-text" search or navigate the content (but only that content that *should* be available to the user) and to index the content as it is updated.

Technical scenario

The technical scenario implemented for this book will enable the Document Search portlet to "crawl" the static content (HTML, flat text) in order to build a searchable index. However, our selection criteria are entered as free-text search terms through a series of configuration parameters. These parameters are as follows:

- ▶ Languages enabled
- ▶ Document types to be indexed
- ▶ Index documents starting from a specific URL
- ▶ Levels of linked documents
- ▶ Documents within a certain domain
- ▶ Number of linked documents to index

These parameters can be used to focus the indexing on a certain subset of documents. However, this requires that the content be pre-organized into a form that can be "consumed" by the portlet (for example the news portlet requires XML in certain format and structure).

The Domino Extended Search capability (also called "Federated Search") will not be used in our technical scenario. However, in an environment where creating an operational data store and/or central index is not practical (for example smaller department-level installations), the extended search's capability of directly querying data sources can provide the same results by accessing separate databases through a single interface. The IBM Lotus Discovery Server will also not be used in our technical scenario.

A WebSphere Commerce Portal is unique in that a commerce search portlet searches and retrieves WebSphere Commerce data from the WebSphere

Commerce instance database either by name-value pairs for products or by text searches to indexed catalog data.

3.3.9 Application Integration::Direct Connection application pattern

The Direct Connection application pattern, as applied to the Electronic Commerce composite pattern, embodies the commerce interaction undertaken by a purchaser in the commerce enable portal solution. A commerce enabled portal solution contains interactions, such as RPQ/RFP and exchange trading, that go beyond what is typically encountered in a B2B scenario. Figure 3-13 shows the matching and selling functions of the commerce enabled portal solution.

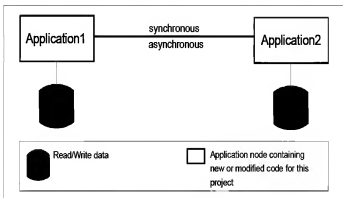


Figure 3-13 Application Integration::Direct Connection

Business and IT drivers

The business and IT drivers are as follows:

- Provide universal access to commerce services
- Time to market
- Reduce total cost of ownership (TCO)

Striving to provide universal access to commerce services and applications is often the primary business driver for choosing this Application pattern.

The primary IT driver for choosing this Application pattern is to quickly extend the user-reach experience without having to modify every individual application to enable its use by additional device types.

Electronic Commerce composite pattern

The Electronic Commerce composite pattern supports the use of direct connection. In fact, commerce access is supported through the use of WebSphere Portal portlets. At this tier, the session data containing the type of device and type of request is known and the properly formatted content can be delivered.

Working example scenario

The working example scenario of this redbook demonstrates access to commerce services provided by WebSphere Commerce.

3.4 Review Runtime patterns

Once the Application pattern has been chosen, the Runtime pattern is selected. The Runtime pattern uses nodes to group functional and operational components. The nodes are interconnected to solve the infrastructure and integration needs of the Application pattern.

In a commerce enabled portal solution, the available Runtime patterns depend largely on the integration requirements, arising from the potential interactions between multiple buyers and suppliers. Development of an e-commerce site is normally iterative with the addition of extra participants and degrees of automation as the site evolves. This evolution should be planned for from the beginning and the site may implement different Application patterns progressing from simpler to more complex patterns with the addition of extra nodes and interactions to implement this.

The deconstruction of the problem into separate nodes provides an initial picture that can be used for a walkthrough to verify the design, initial cost and planning of the selected solution.

The Runtime patterns are graphically illustrated in the following sections. Each pattern will consist of several nodes, describing the function represented on that node. Most patterns will consist of a core set of common nodes, with the addition of one or more nodes unique to that pattern.

Within a typical Runtime pattern, the network is divided into four parts or tiers, as seen in Figure 3-14 on page 94:

- ▶ Outside world
 - Demilitarized Zone (DMZ)
 - Production Zone
- ▶ Internal network

- Intranet zone
- Restricted zone

Candidate nodes include:

- ▶ Browser-based clients
- ▶ Mobile clients
- ▶ Application clients
- ▶ External systems
- ▶ Public Key Infrastructure (PKI) nodes
- ▶ Mobile gateway
- ▶ Protocol firewall
- ▶ Edge server
- ▶ Web portal
- ▶ Transcoding
- ▶ B2B gateway
- ▶ Domain firewall
- ▶ Application server
- ▶ Application database server
- ▶ Directory and security
- ▶ Content management and catalog aggregation
- ▶ Integration hub
- ▶ Mail server
- ▶ Presentation server
- ▶ Collaboration server
- ▶ Personalization server
- ▶ Search and indexing node

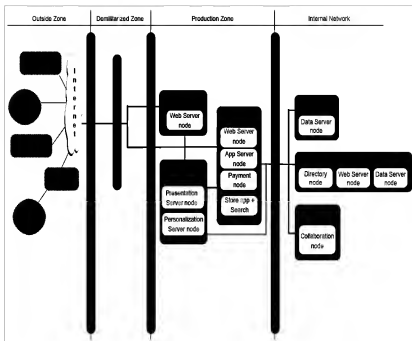


Figure 3-14 Runtime pattern tiers and nodes

For a detailed description of the nodes, refer to Chapter 4, "Business requirements analysis and solution design" on page 99.

3.5 Review Product mappings

In this chapter, we map the logical nodes defined in a Runtime pattern to products. Product mapping identifies the platform, software product name, and version. The IBM e-business infrastructure supports many platforms, including IBM AIX®, IBM OS/400®, IBM OS/390®, Sun Solaris, HP-UX, Linux and Windows NT/2000. We have provided Product mappings for only Windows NT/2000. The most current information on Product mappings can be found on the Patterns for e-business site at:

<http://www.ibm.com/developerworks/patterns/>

The first step in performing a Product mapping is to choose the right platform. The platform chosen should fit into the customer's environment and provide

quality of service, such as scalability and reliability, so that the solution can grow with the e-business. When selecting a platform, some of the factors to consider include:

- Existing infrastructure, systems and platform investments
- Available customer and developer skills
- Customer preference
- Quality of service, such as scalability and reliability

Figure 3-15 on page 95 provides an example Product mapping for the logical Runtime pattern using Windows 2000 Server. It illustrates the software product names and version for the Runtime pattern explained in 4.4.4, "Operational model" on page 134 for the ITSO working example scenario. In the ITSO working example we focus on nodes 1 to 5, as seen in Figure 3-15 on page 95.

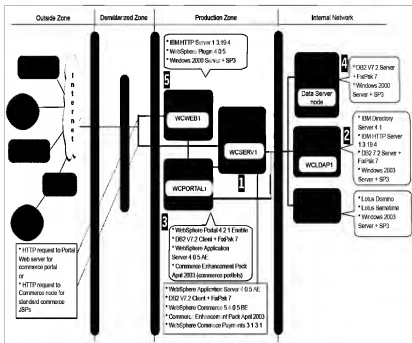


Figure 3-15 Product mapping on Windows 2000

Note: Notice both Web browser and mobile clients can access the WebSphere Commerce node directly (standard JSPs) or through the WebSphere Portal Web server for commerce portlets (see Figure 3-15). For more information on the architecture refer to "WebSphere Commerce Portal component architecture" on page 24.

For information on the Runtime patterns and Product mappings for the Portal composite pattern, refer to *A Portal Composite Pattern Using WebSphere Portal V4.1.2*, SG24-6869 redbook.



Working example scenario: commerce enabled portal



Business requirements analysis and solution design

This chapter describes the ITSO working example business scenario, requirements analysis and solution design for a commerce enabled portal. The focus of this chapter is to provide a clear analysis of the requirements with the creation of use cases and provide a high-level design in the solution design. We choose the fictitious CEP company scenario to illustrate the major phases upon which to build the solution.

The remainder of the chapters will refer to the CEP business scenario. The CEP sells hardware equipment online via the CEP site. We will use the business scenario developed in this chapter to provide a more real-life example of designing and customizing the commerce enabled portal solution.

Important: The requirements and solution design documented are intended as guidance for a real customer engagement. The remaining working example chapters do not implement all aspects of the design.

The chapter is organized into the following sections:

- ▶ Business scenario
- ▶ Requirements
- ▶ Use case model
- ▶ Architecture

4.1 Business scenario

This section describes the business scenario of the fictitious CEP commerce enabled portal solution, and includes the following topics:

- ▶ Initial context, which describes the current business solution of CEP.
- ▶ Business challenges, describing the business drivers for moving towards a commerce enabled portal solution.

Note: Throughout the working example chapters, we will refer to the ITSO sample as the ITSO B2B CEP store.

4.1.1 Initial context

CEP developed a multi-channel business-to-business commerce solution using the WebSphere Commerce Business Edition V5.4. The solution is a customized version of the ToolTech sample store provided by the Commerce Enhancement Pack. It includes the following features:

- ▶ Contract-based purchasing
- ▶ Order management
- ▶ Advanced search
- ▶ Catalog navigation
- ▶ Enhanced buyer-side administrative tasks
- ▶ Enhanced seller-side administrative tasks
- ▶ Enhanced payment integration
- ▶ WAP-enabled
- ▶ Sametime enabled help desk support center

The store supports multi-channel access and the look and feel is customized according to the supported channel.

The customer's security policy enforces specific placements of the system components (see Figure 4-1 on page 101):

- ▶ Databases must be placed within the restricted zone.
- ▶ Business applications, such as the commerce solution, must be placed within the production zone.
- ▶ Development environments are located within the intranet zone.

External users have to pass from the Internet to the demilitarized zone (DMZ) to access the CEP store. The DMZ provides only load balancing and reverse proxy services to enforce security policies. Requests are routed from the Internet to the

DMZ to predefined nodes within the production zone. A set of firewalls secures the access to resources as displayed in Figure 4-1.

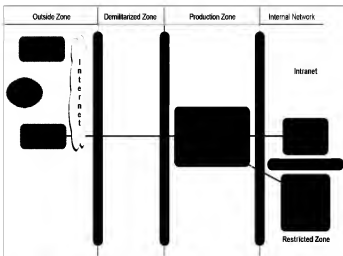


Figure 4-1 Customer's zone topology

4.1.2 Business challenges

After the successful launch of the business-to-business solution, the CEP management wants to leverage and extend its business services. As a result the managers are focusing on an integrated approach or user reach to increase their customer satisfaction, which is a challenging business success factor. In order to evaluate solutions, they have provided the following criteria:

- ▶ Leverage and integrate business services using industry standards.
- ▶ Multi-channel access point for internal and external users (for example, standard Web browser clients, WAP mobile phones, and other mobile device clients).
- ▶ Co-existence with current implemented solutions.
- ▶ Single sign-on.
- ▶ Enhanced commerce capabilities including:
 - Dynamic Context Groups.
 - Rule-based and role-based personalization (not implemented in the ITSO working example).

After several discussions, the managers decide to select a portal-based approach using WebSphere Portal, which supports the mentioned evaluation requirements.

In addition, the Commerce Enhancement Pack provides instructions and information for deploying WebSphere Commerce function into WebSphere Portal. It integrates the advantages, features, and benefits of WebSphere Commerce with WebSphere Portal to enrich the e-commerce functionality provided by the portal.

As an outcome of the evaluation, the management team decides to initiate a project to provide a commerce enabled portal solution for its customers.

4.2 Requirements

This section provides an analysis of the functional and nonfunctional requirements for the business scenario. In several CEP workshops, a set of business user requirements has been identified.

4.2.1 Functional requirements

The CEP company sells hardware and accessories online to retail customers and business customers.

CEP brands and products

The CEP hardware and accessories sold online are categorized into the following brands:

- ▶ Woodworking
- ▶ Power
- ▶ Cordless
- ▶ Accessories

CEP customer types

The CEP store has three distinct customer types:

- ▶ Retail customer

This includes any public Internet shoppers accessing the public CEP site.

- ▶ Business customer

This customer set is predetermined and negotiated by the business users in the form of a contract. For example, a business may negotiate a special price for bundled hardware and accessories. The business would enter into a contract with the CEP site and be able to purchase the contracted goods.

- Business reseller customer

This customer set includes business partners that resell CEP products. CEP grants permissions for business resellers to maintain their product data, including all necessary administrative tasks.

CEP shopping experience requirements

The CEP has several hundred customers. Its main focus is to attract its customers and resellers to the CEP Web site. The catalog contains a well-defined set of goods with special discounts for customers. The CEP site leverages the marketing spots of WebSphere Commerce but the marketing organization would like to enhance the direct marketing based on customer profile data. In summary, the CEP has the following functional requirements:

- Enhanced front-end user experience

The customer logs onto the CEP portal. Based on the customer's personal profile, a pre-configured portal application is available. The customer interacts with the pre-arranged portlets to browse the catalog, product, product items, and its shopping cart. If a customer selects a good and places it into the shopping car, it is refreshed by the system.

- Order-related front-end user experience

Once a customer logs onto the CEP portal, he/she manages his/her orders using the commerce enabled portal and the WebSphere Commerce order subsystem. The order subsystem provides services to modify the current order, including order-related information, to view the order history, including the visualization of the order detail when clicking on an order, and to check out the current order.

- Dynamic data aggregation and visualization

For the customer's convenience, portlets within a portal application should leverage Dynamic Context Groups such that when the user clicks a link in the portlet with a slave relationship, the slave portlets are automatically refreshed by the system.

- Enhanced marketing capabilities

The WebSphere Commerce marketing capabilities are powerful and are included in several JSPs of the CEP site. The WebSphere Portal provides a sophisticated Personalization Engine that is extended to enhance the CEP site. Customers are able to define personalization profiles, which results in dynamic portlet aggregation.

For example, a customer defines product profiles, which result in the dynamic activation of portlets visualizing product-related information from WebSphere Commerce.

CEP business reseller requirements

Due to the growing demand for pervasive or mobile computing, CEP has a need to extend the multi-channel capabilities as follows:

- ▶ Alert services for customer convenience

Currently CEP does not offer business resellers services to send alert messages or SMS messages to customers. In response to the growing demand for customer convenience, CEP decides to provide services to send alert messages and SMS messages. Once a business reseller is logged on, he can edit alert messages and send them to the designated customer using the predefined messaging channels (WebSphere Commerce or SMS).

The customer will have the ability to browse the alert messages.

In the future, CEP extends the SMS solution to allow a customer to send SMS response messages to the WebSphere Commerce node for processing.

- ▶ Multi-channel access using WebSphere Portal

CEP offers multi-channel access paths to leverage the site. WebSphere Commerce supports pervasive computing and CEP wants to leverage this technology with the WebSphere Portal environment. The CEP site is accessible via the portal using standard Web browser clients and WAP-enabled mobile devices.

CEP personalization requirements

The CEP personalization requirements are as follows:

- ▶ Rule-based personalization

Personalization is a key success factor for CEP. Customers appreciate the marketing spots based on CEP's comprehensive marketing reports. WebSphere Portal enhances the personalization using the WebSphere Personalization Engine. A customer can define personal preferences, which are maintained by the portal. Based on the current customer preference data, the Personalization Engine applies the rules and offers the user a set of portlets rendered by the Portal Engine. These portlets access back-end systems including WebSphere Commerce.

- ▶ Role-based personalization

The CEP site offers role-based personalization by leveraging the WebSphere Commerce role-based user model. Registered customers of the CEP site are assigned a set of roles by which they are granted permissions to specific tasks within WebSphere Commerce. CEP wants to offer users specialized portlets using role based personalization.

For example, a business reseller with the role of Buyer Approver is offered an approval portlet once he has logged on and his profile matches the role-based personalization rule.

4.2.2 Nonfunctional requirements

The nonfunctional requirements for a business system address those aspects of the CEP system that can have a profound effect on how that business system is accepted by both the users and the people responsible for supporting the CEP system.

The *nonfunctional* aspects of the CEP business system cover a broad range of themes. The major nonfunctional themes identified are listed below:

- ▶ Performance
- ▶ Scalability
- ▶ Availability (including recoverability and reliability)
- ▶ Maintainability (including flexibility and portability)
- ▶ Security manageability
- ▶ Environmental (including safety)
- ▶ System usability
- ▶ Data integrity (including currency, locality of updating, data retention)

The requirements of the system to be delivered must be understood in each of these areas. All these requirements are presented in a way that facilitates the design and development of the operational model (that is, the computers, networks, and other platforms on which the application will execute and by which it is managed). They also feed into the design of technical and application components. For example, service level requirements may imply component performance requirements.

Together with the functional requirements, they define the baseline against which the business system must be designed. Both sets of requirements are captured during the initial stages of the CEP project.

For convenience, this chapter also includes constraints that the system must conform to or satisfy.

System constraints include:

- ▶ The business constraints that the system must satisfy (for example, geographical location)
- ▶ The technical standards the system must satisfy
- ▶ The technical givens that constrain the system (for example, existing hardware or which DBMS must be used).

Performance

The nonfunctional requirements relating to performance are actually a combination of four separate sets of sub-requirements:

- ▶ Response time requirements
- ▶ Throughput requirements (dynamic volumetric requirements)
- ▶ Utilization requirements
- ▶ Static volumetric requirements

We briefly describe each subset as follows.

Response time requirements

This subset captures those requirements relating to the response time to complete specific processes. It is important to focus on the requirements of the business when setting response time targets, rather than getting seduced by the vision of sub-second IT transaction response. The real driver for the business is its ability to perform a business process in a time that will be acceptable to its customers, not necessarily for each individual system transaction to complete in less than 1 second, interactive or batch, within the target business system. Also, the cost of achieving the response time by purchasing oversized hardware may be inflated for something that is not a user requirement.

In the interactive case, this could be either:

- ▶ The end-to-end response time associated with a specific user-system interaction. For example, the time between a user selecting the **Process** button within a desktop window and the resulting set of data from the associated query being displayed back to the user.
- ▶ The elapsed time to complete a specific business process or transaction, for example the elapsed time to complete the order-entry business process. In this case, it is important to separate out the “think time” from the system time within the overall business process response time target.

In the batch case, this could be the elapsed time to complete the overnight bill run.

Throughput requirements (dynamic volumetric requirements)

This subset captures those requirements relating to the ability of the business system to execute a given number of business or system-related processes within a given unit of time (*design point workload*). For example:

- ▶ The number of account balance enquiries processed per day
- ▶ The number of new orders processed per day
- ▶ The number of telephone call records processed per nightly bill run

This activity will involve an understanding of the frequency of invocation of each of the business processes specified within the process model. A frequency example includes the number of orders per day, along with an understanding of the size of each process and the average number of items per order. The approach taken will depend on the number of processes defined within the overall business system. In practice, many business systems have so many processes that it may be necessary to limit a detailed understanding of the volumetric to a subset. This is typically a combination of high-volume processes (for example, the top 20% most frequently executed processes) plus those that we know will place significant demands on one or more system components. The remaining processes are then managed by making an agreed set of working assumptions.

Utilization requirements

This subset captures those requirements relating to the maximum acceptable loading of the nodes on which the business system is to be implemented when running the design point workload. In some situations, the contract will stipulate that one or more of the delivered system components should not exceed a certain utilization threshold when supporting the design point workload. For example, the network bandwidth utilization must not exceed 20%, or the database server will be no more than 60% utilized.

More commonly, however, this will not be the case. In all situations, the performance architect must size the system such that the overall utilization of the key system components is within the bounds of acceptable behavior when supporting the design point workload. In most cases, it would be unwise to specify a platform configuration that resulted in a design point loading greater than 70%.

Static volumetric requirements

This subset captures those requirements relating to the volumetric for the data entities that exist within the target system that, although relatively static, are likely to have a significant effect on the overall sizing of the target system. Examples include the following:

- ▶ The number of business system users by type
- ▶ The number of different user locations
- ▶ The number of customers
- ▶ The number of customer accounts
- ▶ The number of products

The combination of these four sets of performance sub-requirements are required to properly define the performance baseline to which the system must be designed.

Summary

In summary, the system must be designed to meet the agreed response time requirements, while supporting the design point workload mapped against the given static volumetric baseline, on a system platform that does not exceed the stated utilization.

The nonfunctional requirements deliverable is used for the following reasons:

- ▶ Define requirements and constraints on the IT system. Clear requirements are necessary for a successful project because they define the project's goals. They clarify what is needed and help keep the team focused.
- ▶ As a basis for early system sizing and estimates of cost.
- ▶ To assess the viability of the proposed IT system.
- ▶ To drive the design of the operational models. Nonfunctional requirements are frequently the most important determining factor of the architecture. Two systems with the same use cases but with very different nonfunctional requirements need very different architectures.
- ▶ As an input (of SLRs and other nonfunctional requirements) to component design.

The performance and scalability-related nonfunctional requirements (NFRs) provide the baseline against which the subsequent performance engineering activities are scoped and executed.

Mistakes made now will be paid for dearly in the future, through the failure of the target system (for example, meet the specified response time targets), the scalability demands of the client's workload, the specified availability targets, etc. In all such cases, the delivered system will fail to meet the needs or expectations of the business.

Impossible or unreasonable requirements should be rejected at this stage. If the client is unwilling to dilute these requirements, then the project needs to make a judgment as to whether to continue or not. If the targets are merely challenging, then sufficient effort should be built into the overall project plan to address them.

For the CEP commerce enabled portal solution, the following nonfunctional requirements are identified:

- ▶ Performance
- ▶ Availability
- ▶ System constraints
- ▶ Volumetric

Performance

Table 4-1 provides a set of generic response time bands for the most significant user-system interaction within the different classes of transaction present in the CEP business system.

Table 4-1 Response time requirements

Business Transaction Complexity	High Frequency (e.g. > 100 times per day)	Medium Frequency (e.g. > 10 times per day but < 100 times per day)	Low Frequency (e.g. < 10 times per day)
Simple Transaction	1 - 2	2 - 3	3 - 4
Medium Transaction	3 - 5	4 - 7	5 - 10
Complex Transaction	6 - 10	8 - 15	11 - 20
Very Complex Transaction	11 - 20	16 - 30	21 - 40
Other	> 20	> 30	> 40

Note: Performance requirements for collections of uses cases can be represented in a similar matrix form.

Availability

Availability is frequently an important service level requirement. Table 4-2 provides an overview of availability specification. Availability requirements vary by the CEP use cases; each row represents a collection of use cases with common availability requirements.

Table 4-2 Availability requirements

Use Cases	Service hour	Fallback plan	Availability Req VH/H/M *	Recovery requirement	Number of outages acceptable
UCF1 UCF2 UCF3 UCF4 UCF5 UCF6 UCF10	almost 24 hours	Use native WebSphere Commerce	VH	1 hour recovery	1/month

Use Cases	Service hour	Fallback plan	Availability Req VH/H/M *	Recovery requirement	Number of outages acceptable
UCF7 UCF8	08:00 - 18:00	Use native WebSphere Commerce	M	Next day	2/year
UCF9	08:00 - 18:00	Use native WebSphere Commerce	H	1/2 hour recovery	1/year
UCS1 UCS2	20:00 - 24:00		M	Next day	1/month
UCS3	almost 24 hours		VH	1/2 hour	1/year
UCA1 UCA2 UCA3	20:00 - 24:00		M	Next day	2/month

* For the purpose of this book we assume the availability of 24x7, but we are not providing an exact solution needed to achieve this requirement.

System constraints

Table 4-3 lists known system constraints for the CEP production environment and associated prerequisites including mandatory standards of component structure.

Table 4-3 System constraints for CEP solution

Constraint ID	Comment
SC01	Access control and security <ul style="list-style-type: none"> – WebSphere security – LDAP Access Control Lists
SC02	Database <ul style="list-style-type: none"> – DB2® on Windows 2000 Advanced Server
SC03	SMS gateway <ul style="list-style-type: none"> – WebSphere Everyplace Wireless Gateway for AIX
SC04	Production servers <ul style="list-style-type: none"> – Intel® Pentium® III, 1 GHz PCs, 1 GB memory and software (Windows 2000 etc.)
SC05	Development workstations <ul style="list-style-type: none"> – Intel Pentium III, 1.8 GHz PCs, 2 GB memory and software (Windows 2000 etc.)

Constraint ID	Comment
SC06	Client browsers <ul style="list-style-type: none"> – Internet Explorer >= 5.5 or higher – Netscape Navigator >= 4.7 or higher – WAP-enabled devices
SC07	Network <ul style="list-style-type: none"> – TCP/IP and Ethernet
SC08	Separation of End User Component, Application Component, and Data Component <ul style="list-style-type: none"> – Each component can be amended without dependence on amendments to any other components.
SC09	Characteristics of Application Component <ul style="list-style-type: none"> – Performs business function and validation – Portable to other hardware and software platforms without amendment – Each application unit of work can be executed in isolation of other application units of work – Application units of work are separate from data accesses
SC10	Characteristics of Data Component <ul style="list-style-type: none"> – Relational – Maintain integrity at all times, whatever the application units of work and whatever platform they are running on – Full recoverability at the level of the application unit of work – Data accesses always are called from an application unit of work – Zero data duplication as a goal compromised only for performance, recoverability, integrity, or security reasons – Portable to other relational databases on other software and hardware platforms

Note: We chose bigger development machines since they are running the development tools to.

Volumetric

Table 4-4 lists the volumetric nonfunctional requirements per use case.

Table 4-4 Volumetric requirements

Use Cases	Information	Volume Information	Sizing Information
UCF1	Request to register a customer	Small amount of data	20 registrations per day 10000 customer accounts active
UCF2	Navigate CEP catalog and products including product items	Large amount of data	200 concurrent users per hour 13000 products
UCF3 UCF4	Maintain shopping cart and process orders	Medium amount of data	6500 daily orders
UCF5	Navigate and view alert messages	N/A	approximately 200 alerts per day
UCF6	Rule based personalization	Medium amount of data	320 marketing spots
UCF7	Approve user registration	Small amount of data	20 per day
UCF8	View, check and approve a customer order	Medium amount of data	6500 per day
UCF9	Request to alert a customer about an order state	Relatively small amount of data	approximately 200 requests per day
UCF10	Role-based personalization	Small amount of data	
UCS1	Deploy a CEP store including catalog data and site information	Large amount of catalog data Medium amount of site data including JSPs High amount of data for images	13000 products including product items 3 stores with 10-20 categories each 3 sets of JSPs for customer profile
UCS2	Deploy a portal application	Small amount of data	

Use Cases	Information	Volume Information	Sizing Information
UCS3	SMS message to alert a customer	Small amount of data limited by the SMS protocol	
UCA1 UCA2 UCA3	Add, edit, and delete portlet in WebSphere Portal	N/A	

4.3 Use case model

This section underscores the functional requirements of the CEP site represented by the use case model. The model can use graphical symbols and text to specify how users (actors) in specific roles will use the system (modeled by use cases). The textual descriptions describing the use cases are from the actor's point of view; they do not describe how the system works internally or the internal structure or mechanisms.

A whole set of use cases for the CEP site would cover business, administration, and system area. Therefore, only some front-end user experience, administration, and system use cases are addressed in this section.

4.3.1 Use case overview

This section describes the CEP actors and summarizes the use cases.

Actors defined

The actors are people or systems outside of the WebSphere Commerce Server that interact with the "system" (CEP site). This section identifies primary and secondary actors, where a primary actor is one that initiate a use case, and a secondary actor is one that is involved in one or more of the steps in the use case but does not initiate it. Figure 4-2 on page 114 depicts actors modeled for the CEP site.

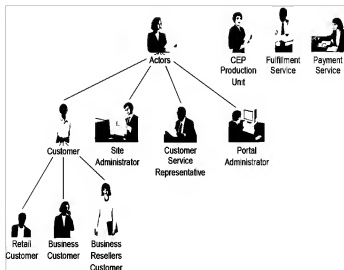


Figure 4-2 CEP site actors

Primary actors

The primary actors are as follows:

- ▶ Customer
- ▶ Site Administrator
- ▶ Customer Service Representative (CSR)
- ▶ Portal Administrator

Secondary actors

The secondary actors are as follows:

- ▶ CEP production unit
- ▶ Fulfillment services
- ▶ Payment services

Summary of use cases

This section lists the use cases, describes the goals of each use case, and shows the primary actors. As previously noted, only front-end user experience, system, and administration use cases are elaborated.

The front-end use cases can be found in Table 4-5 on page 115. System use cases can be found in Table 4-6 on page 115. Administration use cases can be found in Table 4-7 on page 116.

Table 4-5 Front-end user experience use cases

Use case ID	Use case name	Goal in context	Primary actor
UCF1	Register customer	Register customer in LDAP including associated organization profile.	Customer
UCF2	Navigate through catalog	Navigate through catalog and dynamically update product portlet using dynamic context.	Customer
UCF3	Maintain shopping cart	Add/Delete products to/from shopping cart.	Customer
UCF4	Process orders	Display current order information and proceed with booking. Update order history and show order detail when historical order data is selected.	Customer
UCF5	View alert messages	Registered user can view alert messages raised by the customer representative.	Customer
UCF6	Rule-based personalization	Show portlet on rule-based customer preferences.	Customer
UCF7	Registration approval	To approve an user registration request.	Site Administrator
UCF8	Order approval	To view, check and approve an order request.	Site Administrator
UCF9	Order alert	To alert a customer about an order request state.	Site Administrator
UCF10	Role-based personalization	Show portlet based on role-based preferences.	Site Administrator

Table 4-6 System use cases

Use case ID	Use case name	Goal in context	Primary actor
UCS1	Publish store	Publish the CEP store to the production server	System administrator or None

Use case ID	Use case name	Goal in context	Primary actor
UCS2	Publish portal application	Publish the portal application to the production server	Portal administrator or None
UCS3	Send SMS	Sending a SMS message to the designated SMS owner	

Table 4-7 Administration use cases

Use case ID	Use case name	Goal in context	Primary actor
UCA1	Add portlet	WebSphere Commerce portlet is deployed on the productive server	Portal administrator
UCA2	Edit portlet	WebSphere Commerce portlet is edited on the productive server	Portal administrator
UCA3	Delete portlet	WebSphere Commerce portlet is deleted on the productive server	Portal administrator

4.3.2 Front-end experience use cases detail

This section provides details about each of the front-end experience use cases that have been designed for customer interaction through the commerce enabled portal. These use cases cover a standard shopping flow from the customer perspective, approval use cases for site administrators, and personalization.

Table 4-8 UCF1: Register customer

Use Case ID: name	UCF1: Register Customer
Description	This is the primary use case to register a customer to access the CEP commerce enabled portal.
Precondition	None
Primary actor	Customer
Secondary actor	Site Administrator

Use Case ID: name	UCF1: Register Customer
Main Scenario	<ol style="list-style-type: none"> 1. Actor fills out the registration form. <ol style="list-style-type: none"> a. Actor uses the WebSphere Portal registration form. b. Actor uses the WebSphere Commerce registration portlet. c. Actor uses the WebSphere Commerce registration form of the CEP site. 2. Actor submits the registration form. 3. System receives the registration request and checks the integrity of data. 4. System creates an LDAP entry according to the registration profile. 5. Use case ends successfully and actor is logged in.
Alternatives	<ol style="list-style-type: none"> 1. Register actor using the WebSphere Commerce administration console. 2. Create an actor LDAP entry using the IBM Directory Management Tool. 3. LDIF import. 4. JNDI update.

Note: Users added to the LDAP directory are not synchronized with the WebSphere Commerce database until login. A mapping of fields to synchronize is maintained in the WebSphere Commerce `ldapentry.xml` file.

Table 4-9 UCF2: Navigate through catalog

Use Case ID: name	UCF2: Navigate through catalog
Description	This is a primary use case to navigate through the catalog of the CEP store. It covers the catalog, product and item navigation using dynamic context.
Precondition	Actor is logged in via WebSphere Portal.
Primary actor	Customer.
Secondary actor	None.

Use Case ID: name	UCF2: Navigate through catalog
Main Scenario	<ol style="list-style-type: none"> 1. Actor browses CEP catalog. 2. System updates product display when the actor selects a catalog category. 3. Actor selects product or product item. 4. System updates the product item display. 5. Actor adds the product item to the shopping cart. 6. System updates the shopping cart. 7. Use case ends successfully.
Alternatives	1. Actor uses standard CEP store front end.

Table 4-10 UCF3: Maintain shopping cart

Use Case ID: name	UCF3: Maintain shopping cart
Description	This is a primary use case to maintain the shopping cart. It covers the visualization of the shopping cart and shopping cart item manipulation.
Precondition	Actor is logged in via WebSphere Portal.
Primary actor	Customer.
Secondary actor	None.
Main Scenario	<ol style="list-style-type: none"> 1. Actor modifies shopping cart elements. 2. System updates shopping cart. 3. Use case ends successfully.
Alternatives	1. Actor uses standard CEP store front end.

Table 4-11 UCF4: Process orders

Use Case ID: name	UCF4: Process orders
Description	This is a primary use case to handle the order-related tasks of a customer. It covers the order processing and management of the order history and order history detail.
Precondition	Actor is logged in.
Primary actor	Customer.
Secondary actor	None.

Use Case ID: name	UCF4: Process orders
Main Scenario	<ol style="list-style-type: none"> 1. Actor submits current order for processing: <ol style="list-style-type: none"> a. System checks integrity of order-related data. b. System updates current order and order history. 2. Actor navigates through order history. <ol style="list-style-type: none"> a. System updates order history detail based on the selected order. 3. Use case ends successfully.
Alternatives	<ol style="list-style-type: none"> 1. Actor uses standard CEP store front end.

Table 4-12 UCF5: View alert messages

Use Case ID: name	UCF5: View alert messages
Description	This is a primary use case to view customer's alert messages.
Precondition	None.
Primary actor	Customer.
Secondary actor	Site Administrator.
Main Scenario	<ol style="list-style-type: none"> 1. Actor logs in and views alert messages. 2. System updates alert message display. 3. User case ends successfully.
Alternatives	None.

Table 4-13 UCF6: Rule-based personalization

Use Case ID: name	UCF6: Rule-based personalization
Description	This is a primary use case to personalize the commerce enabled portal solution. It covers the preference definition and rule-based Personalization Engine to provide personalized content.
Precondition	Portal administrator publishes personalized portlets.
Primary actor	Customer.
Secondary actor	Portal Administrator.

Use Case ID: name	UCF6: Rule-based personalization
Main Scenario	<ol style="list-style-type: none"> 1. Actor enters preference data. 2. System creates resource object stored in LDAP. 3. System activates rule-based personalization based on Actors preference data. 4. Actor views personalized content. 5. Use case ends successfully.
Alternatives	None

Table 4-14 UCF7: Registration approval

Use Case ID: name	UCF7: Registration approval
Description	This is a secondary use case to approve registration requests. It covers the approval tasks to view and approve/reject the registration request of a customer.
Precondition	Customer initiated a registration request and site administrator is logged in.
Primary actor	None.
Secondary actor	Site administrator.
Main Scenario	<ol style="list-style-type: none"> 1. Actor views the registration request(s). 2. Actor approves/rejects a selected registration request. 3. System processes the approval or rejection 4. Use case ends successfully.
Alternatives	<ol style="list-style-type: none"> 1. Actor uses the WebSphere Commerce administration tools.

Table 4-15 UCF8: Order approval

Use Case ID: name	UCF8: Order approval
Description	This is primary use case to approve orders that cannot be automatically processed by the system. It covers browsing the orders and/or approving/rejecting them.
Precondition	None.
Primary actor	Site administrator.
Secondary actor	None.

Use Case ID: name	UCF8: Order approval
Main Scenario	<ol style="list-style-type: none"> 1. Actor views orders to be approved. 2. Actor approves/rejects selected order. 3. System performs approval/reject request. 4. Use case ends successfully.
Alternatives	<ol style="list-style-type: none"> 1. Actor uses the WebSphere Commerce administration tools.

Table 4-16 UCF9: Order alert

Use Case ID: name	UCF9: Order alert
Description	This a primary use case to alert a customer about an order. It covers the generation of a standard alert message within the CEP store.
Precondition	None.
Primary actor	Site Administrator.
Secondary actor	None.
Main Scenario	<ol style="list-style-type: none"> 1. Actor creates an alert message for a registered customer. 2. System delivers alert message. 3. Use case ends successfully.
Alternatives	None.

Table 4-17 UCF10: Role-based personalization

Use Case ID: name	UCF10: Role-based personalization
Description	This is a primary use case to provide role-based personalization. It covers the definition of role-based preferences and the activation of the associated personalized content.
Precondition	Portal administrator has published portlets.
Primary actor	Customer.
Secondary actor	Portal administrator.

Use Case ID: name	UCF10: Role-based personalization
Main Scenario	<ol style="list-style-type: none"> 1. Actor specifies role-based personalization preferences. 2. Actor submits personalization request. 3. System stores personalization resource object within LDAP. 4. System renders portal including role-based personalized content. 5. Use case ends successfully.
Alternatives	None.

4.3.3 System use cases detail

This section provides detail for each of the system use cases designed for the publishing of commerce enabled portals. It includes the system use case to send SMS messages.

Table 4-18 UCS1: Publish store

Use Case ID: name	UCS1: Publish store
Description	This is a primary use case to publish the CEP store. It covers the steps of using the WebSphere Commerce administrative tools.
Precondition	A store to publish.
Primary actor	Site administrator.
Secondary actor	None.
Main Scenario	<ol style="list-style-type: none"> 1. Actor starts the WebSphere Commerce store services. 2. Actor logs in and initiates the publishing process. 3. System publishes the store. 4. Use case ends successfully.
Alternatives	None.

Table 4-19 UCS2: Publish portal application

Use Case ID: name	UCS2: Publish portal application
Description	This is a primary use case to publish a commerce portal solution for WebSphere Portal. It covers the steps to publish the portal application and to configure the access rights.

Use Case ID: name	UCS2: Publish portal application
Precondition	Portal application to publish.
Primary actor	Portal administrator.
Secondary actor	None.
Main Scenario	<ol style="list-style-type: none"> 1. Portal administrator initiates the publishing process for the portal application. 2. System publishes the portal application. 3. Portal administrator defines access rights. 4. Use case ends successfully.
Alternatives	None

Table 4-20 UCS3: Send SMS

Use Case ID: name	UCS3: Send SMS
Description	This a primary use case to send a SMS message to a customer. It covers the definition of the SMS message and its delivery process.
Precondition	None.
Primary actor	Site administrator.
Secondary actor	None.
Main Scenario	<ol style="list-style-type: none"> 1. Actor enters data for SMS delivery. 2. Actor submits SMS request. 3. System create SMS message and delivers it. 4. Use case ends successfully.
Alternatives	None.

4.3.4 Administration use cases detail

This section provides detail for each of the administration use cases designed for the maintenance of portlets.

Table 4-21 UCA1: Add portlet

Use Case ID: name	UCA1: Add portlet
Description	This is a primary task to add a portlet to the commerce enabled portal solution. It covers the tasks to define and publish the portlet.
Precondition	None.
Primary actor	Portal administrator.
Secondary actor	None.
Main Scenario	<ol style="list-style-type: none"> 1. Actor selects portlet to publish. 2. Actor initiates publish request. 3. System publishes portlet. 4. Use case ends successfully.
Alternatives	None.

Table 4-22 UCA2: Edit portlet

Use Case ID: name	UCA1: Edit portlet
Description	This is a primary task to edit a portlet of the commerce enabled portal solution. It covers the tasks to edit and maintain the portlet.
Precondition	None.
Primary actor	Portal administrator.
Secondary actor	None.
Main Scenario	<ol style="list-style-type: none"> 1. Actor selects portlet to edit. 2. Actor edits portlet definitions and submits an update request. 3. System update portlet definition. 4. Use case ends successfully.
Alternatives	None.

Table 4-23 UCA3: Delete portlet

Use Case ID: name	UCA1: Delete portlet
Description	This is a primary task to delete a portlet from the commerce enabled portal solution. It covers the tasks to delete a published portlet.

Use Case ID: name	UCA1: Delete portlet
Precondition	None.
Primary actor	Portal administrator.
Secondary actor	None.
Main Scenario	<ol style="list-style-type: none"> 1. Actor selects portlet to delete. 2. Actor deletes portlet. 3. System deletes portlet. 4. Use case ends successfully.
Alternatives	None.

4.4 Architecture

At the enterprise level, the architecture helps communicate to the sponsor and all stakeholders an understanding of the overall future directions for the IT environment. This understanding will help management decision making about major strategic IT investment, acquisitions, and sourcing. It provides a high-level shared vision of the architecture and scopes of potential future IT systems.

The architecture section is used to provide an understanding of the following topics:

1. Communicate to the sponsor and external stakeholders a conceptual understanding of the intended IT system.
2. Provide a high-level shared vision of the architecture and scope of the proposed IT system for the development teams.
3. Explore and evaluate alternative architectural options.
4. Enable early recognition and validation of the implications of the architectural approach.
5. Facilitate effective communication between different communities of stakeholders and developers.
6. Facilitate orientation for new people who join the project.

4.4.1 Architecture overview

An architecture overview diagram in Figure 4-3 on page 126 represents the governing ideas and candidate building blocks of a commerce enabled portal CEP site system. It provides an overview of the main conceptual elements and

relationships in an architecture, including candidate subsystems, components, nodes, connections, data stores, users, and external systems.

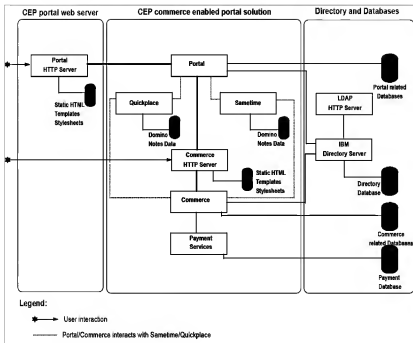


Figure 4-3 CEP architectural overview diagram

Note: The remote Portal HTTP Server is only used by the WebSphere Portal server in our design.

The main purpose of the architecture overview diagram is to communicate a simple, brief, clear, and understandable overview of the target IT system, providing a conceptual view of the components and operational environment as seen in Figure 4-3.

Key concepts

The following are the key IT system level architectural concepts and describes each subsystem in more detail:

- **Portal HTTP Server**

This subsystem enables users to access the CEP commerce enabled portal solution while using WebSphere Portal functionality. User requests are passed to the portal services for processing.

- **Commerce Services**

This subsystem enables users to use common WebSphere Commerce functionality, such as browsing the catalog, maintaining the shopping cart, placing an order, etc. This service includes registration and profile management enabling users to register in a CEP site and work with their profile information. The user registry is synchronized with the IBM Directory Server where user profiles are stored. This data is used for authentication. Included in this subsystem are enhanced role-based administration services. In addition, the WebSphere Commerce services offers interfaces for portal integration and Web services support.

- **Portal Services**

This subsystem enables users to use common WebSphere Portal functionality, such as single sign-on, customization of skins and themes, access to portlets, etc. This service includes registration and profile management enabling users to register in CEP site and work with their profile information. The Personalization Engine allows the definition of rule and role-based profiles to personalize the user front-end experience. The portal services includes a variety of portlets to access business services among various applications. In advance, it is enabled for multi-channels.

- **Quickplace Services**

Collaborative workspaces are handled by the Quickplace services. The services are accessible using portlets within the portal services or the CEP commerce site.

- **Sametime Services**

This subsystem enables users to leverage instant messaging capabilities such as online-help, chat, electronic meetings, etc. It is a base service for CEP customer care and online help support. The messaging capabilities are leveraged within the WebSphere Portal to provide users services such as enhanced search, online help, etc. (for example, Sametime).

- **Payment Services**

This subsystem serves all payment request from WebSphere Commerce.

- **Directory Services**

This subsystem enables the enterprise directory services for CEP. It is able to link with the existing applications through a broad range of interfaces. In addition, management tools provides services to administrate the directory entries and directory structure. This subsystem is the master for single sign-on while holding all necessary user profile data for authentication. The management services are browser based or native Java clients.

- **Database Services**

This subsystem provides services to store, manipulate and retrieve data from relational databases. Several databases are being used by the applications.

4.4.2 Architectural decisions

This section documents important decisions about any aspect of the architecture, including the structure of the system, the provision and allocation of function, the contextual fitness of the system, and adherence to standards.

An architecture is understood partly through the record of the important decisions made during its development. A well-documented architecture includes its own justification and evaluation criteria. The justification and evaluation criteria may be recorded alongside the decision or, at least in part, by reference to more generally applicable principles, policies and guidelines, which are found in other documentation or in external references.

The purpose of the architectural decision section is as follows:

- Provide a single place to find important architectural decisions.
- Make explicit the rationale and justification of architectural decisions.
- Preserve design integrity in the provision of functionality and its allocation to system components.
- Ensure that the architecture is extensible and can support an evolving system.
- Provide a reference of documented decisions for new people who join the project.
- Avoid unnecessary reconsideration of the same issues.

For the CEP working example, we have identified the following architectural decisions:

- AD01: Commerce integration approach (see Table 4-24 on page 129)
- AD02: Portlet calls WebSphere Commerce URL controller command (see Table 4-25 on page 129)

- AD03: Enterprise Directory Server (see Table 4-26 on page 130)
- AD04: Database are kept in the restricted zone (see Table 4-27 on page 131)
- AD05: High availability and scalability (see Table 4-28 on page 131)

Table 4-24 AD01: Commerce integration approach

Architectural Decision ID	AD01
Architectural Decision	Integration approach of CEP commerce application and the WebSphere Portal.
Problem Statement	The CEP commerce application is a productive environment that serves hundreds of customers. The impact of the portal integration must be minimized such that the given front-end experience, the current store front end, and the implementation of business logic is preserved.
Assumptions	Minimize additional development of business logic.
Motivation	Reduce development effort and complexity. Time to market by enabling the commerce services through the portal channel.
Alternatives	<ol style="list-style-type: none"> 1. Leverage the Commerce Enhancement Pack by modifying the CEP commerce JavaServer Pages for the customized portal front-end experience (HTML, WAP WML). 2. Implement Web services for commerce business services and develop portlets to interact with the WebSphere Commerce Server. Define XML output to render the markup language by the WebSphere Portal.
Decision	The first alternative was selected to minimize the development efforts and to reach the time to market goal.

Table 4-25 AD02: Portlet calls WebSphere Commerce URL controller command

Architectural Decision ID	AD02
Architectural Decision	A portlet calls the WebSphere Commerce URL controller command using HTTP or HTTPS.
Problem Statement	The WebSphere Commerce Server controller commands can be called via HTTP/HTTPS or Java Remote Method Invocation calls. Remote Method Invocation has the disadvantage that the firewalls have to be configured. In addition, a port range has to be defined.

Architectural Decision ID	AD02
Assumptions	None.
Motivation	Reduce firewall configuration and force usage of standard protocols.
Alternatives	<ol style="list-style-type: none"> 1. Use Commerce Enhancement Pack software stack to call WebSphere Commerce URL controller commands using HTTP/HTTPS. 2. Develop Remote Method Invocation stubs to directly call WebSphere Commerce Java commands. 3. Implement Web services for commerce business services including SOAP messages and WSDL files.
Decision	The first alternative was selected due to time to market constraints.

Table 4-26 AD03: Enterprise Directory Server

Architectural Decision ID	AD03
Architectural Decision	User profile data and organization data are kept in a centralized enterprise directory service
Problem Statement	Central administration of user profile is business critical for CEP. The data must be manageable in a distributed environment from designated management nodes.
Assumptions	Applications support the Lightweight Directory Access Protocol (LDAP).
Motivation	The enterprise directory server is used as a centralized information repository to support information sharing among various applications
Alternatives	<ol style="list-style-type: none"> 1. IBM Directory Server V4.1 is a standard based enterprise directory server which can operate with multiple applications. 2. Use other third-party directory services with the need for customization.
Decision	The first alternative was selected due to ease of integration. Especially, it is free of charge.

Table 4-27 AD04: Database are kept in the restricted zone

Architectural Decision ID	AD04
Architectural Decision	All databases are kept in the restricted zone due to the CEP security policies
Problem Statement	Mission-critical data assets are stored within the CEP databases. A possible attack from the Internet could result in database manipulations with unknown side effects.
Assumptions	Firewalls are set up and configured as defined by the CEP security policies.
Motivation	Data protection and access control. Load balancing with the ability to define high available database clusters.
Alternatives	None
Decision	Databases are placed in the restricted zone.

Table 4-28 AD05: High availability and scalability

Architectural Decision ID	AD05
Architectural Decision	CEP commerce enabled portal solution is designed for high availability and scalability.
Problem Statement	The CEP portal application is mission critical and is designed to handle all Internet, extranet and intranet requests. The load cannot be estimated due to the unknown growth of customer base. The system must be suitable to scale according to the growth rate of the business.
Assumptions	<ol style="list-style-type: none"> 1. WebSphere Portal supports three-tier approach. 2. WebSphere Commerce Server node is able to handle the load. It is high available and scalable.
Motivation	Separation of the HTTP server, portal server and database server allows CEP to reach scalability and high availability requirements. The load is balanced among all involved components.
Alternatives	<ol style="list-style-type: none"> 1. WebSphere Portal follows the three-tier approach: HTTP server, portal server and database server on different physical nodes. 2. WebSphere Portal installed on a single machine with high availability features.
Decision	The first alternative was selected.

4.4.3 Component model

The component model describes the entire hierarchy of components in terms of responsibilities, interfaces, (static) relationships, and the way they collaborate to deliver required functionality.

A component is a relatively independent part of a system. It is characterized by its responsibilities and eventually by the interface(s) it offers. Components can be decomposed into smaller components or composed into larger components. A component (as used here) is a very general concept and is not restricted to any particular technology.

Most components are software, though sometimes it is convenient to include a hardware and software combination in the component model. A component can be a software subsystem, a program module, a collection of classes (for example, all the classes dealing with orders), a program (for example, one that performs event notification), a part of a product (for example, DB2 or MQSeries®, now called WebSphere MQ), or a hardware device (for example, a scanner). Some components are already available as off-the-shelf products or from a previous project. Some are designed and built on the project.

Components can in principle be of any size. However, as the purpose of the deliverable is architectural, the decomposition tends to stop at a level suitable for allocating work to teams. Detailed work within a team is handled in design deliverables.

A component model often evolves through several stages taking into account successive system distribution, the use of specific products, the choice of middleware, and other technologies.

The component model is used as follows:

- ▶ Describe the high-level structure of the software.
- ▶ Describe precisely the responsibilities, relationships, and interactions of components.
- ▶ Document how application/technical parts of the system are related.
- ▶ Specify how existing, acquired, and developed components are related.
- ▶ Define the components that have to be placed on the operational model, that is, that have to execute and be managed on the target platforms.
- ▶ Help organize the development project.
- ▶ Reduce complexity through the encapsulation offered by a component. Since a component is larger than a class or procedure, partitioning into components simplifies design and development of large and complex systems.

- ▶ Serve as the unit of work allocation. The encapsulation provided by the components reduces the interaction between development groups and makes the overall management easier.
- ▶ Define precisely the boundaries between reusable parts of the system and between development teams.

The component description is mainly text and follows a template such as the following:

- ▶ Responsibilities

A description of the responsibilities of the component. This description will become more detailed as the project progresses. Eventually it will become the definition of the operations offered by the component. These operations can be organized into one or more contracts (or interfaces) offered by the component. Their detail may be put into a separate interface or into the application programming interface deliverable.

- ▶ Required service levels

- General

Information that does not fit into any other category.

- Users and presentation:

If relevant, the number of users of this component, with usage profiles. In many cases, the “user” may be another component. If presentation style is not important for particular user groups, it need not be included here.

- Performance and capacity

The rate at which each interface will be called (average and peak) and the average size of messages. The latter should always be in the context of a particular collaboration. Also include the size of the data, if any, associated with this component, allowing for the particular technology used for the persistence mechanism.

- Availability

The required availability profile, both in terms of hours per day and days per week, together with the percent unavailability sustainable and the required recovery time. In most cases, this will be a shared requirement among many components.

- Security

Any special security requirements for this component or its data.

- Systems management

Any special requirements in terms of change control, for example, or other aspect of systems management.

- ▶ Reasonableness and risk
An assessment of the risk associated with this design, together with a mitigation strategy if the risk is high.
- ▶ Design rationale
A description of the rationale for the key design decisions that were made during the design of this component.
- ▶ Implementation approach
A reference to the deliverable, in those cases where a component selection has taken place, describing the mapping to the product(s). Otherwise, indicate the particular implementation/technology strategy chosen.

We refer to the existing CEP component model as it accommodates the new application. Therefore, only a test of "fit," and possibly some refinements or extensions to the model, is needed.

4.4.4 Operational model

The operational model represents a network of computer systems, their associated peripherals and the systems software, middleware, and application software that they run. In general, it includes the following:

- ▶ One or more diagrams that show the topology and geographic distribution of the system, the node definitions and network connections as well as how users and external systems interact with the commerce enabled portal being developed.
- ▶ A detailed description of each node.
- ▶ A mapping matrix of deployment units to nodes. Each deployment unit is a convenient grouping of components from the software architecture.
- ▶ A description of the system management strategy.
- ▶ A description of middleware services and products and the key middleware choices.
- ▶ Descriptions of walkthroughs, which describe the flow of a business activity from a user all the way through the system and back to the user.

Depending on the stage of analysis and design, nodes and connections may be conceptual, specified, or actual physical computer systems:

- ▶ *Conceptual* corresponds to an early stage of design. Conceptual nodes ignore many technological limitations and focus on application software components, deferring treatment of middleware and other software.

- ▶ *Specified* refers to a detailed specification of a computer platform or network. Technological limitations are fully taken into account but the detailed choice of technology is not made.
- ▶ *Physical* refers to the specific types of computers, networks, and software that make up the system.

Generally an operational model develops from conceptual to specified to physical. Depending on the complexity of the problem and the starting point, it may not be necessary to go through all three stages. At any one time, different parts of the description may be at different levels.

The operational model is the main description of the systems architecture. Without an operational model, or something very similar, it is hard to see how a systems architecture of any complexity can be designed and developed. The most obvious consequence is that some nonfunctional requirements will not be met, and this will not be realized until late in the project.

The commerce enabled portal solution covers the conceptual, specified, and physical model. It is used to implement the runtime environment as described in Chapter 5, "Implement the runtime environment" on page 165.

Conceptual model

The conceptual model of the CEP commerce enabled portal solution sketches all required nodes and provides detailed information at a conceptual level. It is a snapshot of the current CEP environment extended by nodes to accomplish the portal solution. The level of abstraction affects the description of the conceptual nodes, as for example:

- ▶ Numbers and location of users.
- ▶ Location and nature of interfaces to external systems.
- ▶ Business-related deployment units and their operational requirements (such as resources, availability and security).
- ▶ Geographical location of these deployment units and the associated decisions about their replication, distribution, etc.
- ▶ Identifying the nodes and connections required.

To focus on the CEP commerce enabled solution, we focus on the production zone and internal network excluding the outside, DMZ, and firewall nodes. In summary, we describe the following conceptual nodes:

- ▶ Web Server
- ▶ App Server
- ▶ Payment
- ▶ Directory

- Collaboration
- Instant Messaging
- Portal Development
- Commerce Development

Figure 4-4 shows the conceptual model for CEP across the zones.

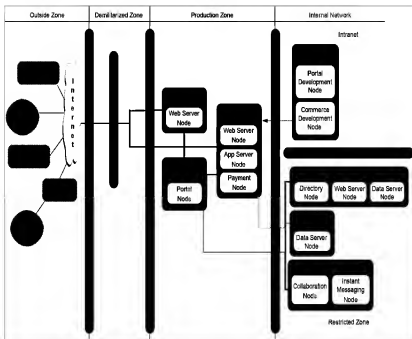


Figure 4-4 Conceptual model of the commerce enabled portal solution

The CEP zone topology is the following:

- Outside zone

It contains the public key infrastructure, user and domain name services that access the CEP site from the internet. it is not under control of cep.

- demilitarized zone

This zones restricts the access to/from internal services of CEP. It contains dispatcher and proxy nodes for load balancing and reverse proxy services. It provides the gateway to authenticate a user and to pass verified requests to the production zone.

► Production zone

The production zone provides presentation and business logic services. This is the place where the current WebSphere Commerce CEP site is hosted. It is accessible from the demilitarized zone or the internal network.

► Internal network

– Intranet

The corporate CEP intranet provides a development domain. Within the developer domain a set of development services are provided to develop CEP solutions.

– Restricted zone

The restricted zone contains all mission-critical CEP assets that have to be protected as stated by the CEP security policy.

The following tables describe the conceptual nodes in detail.

- Web Server node (see Table 4-29)
- Application Server node (see Table 4-30 on page 138)
- Payment node (see Table 4-31 on page 139)
- Directory Server node (see Table 4-32 on page 140)
- Data Server node (see Table 4-33 on page 141)
- Collaboration node (see Table 4-34 on page 141)
- Instant Messaging node (see Table 4-35 on page 142)
- Portal Development node (see Table 4-36 on page 143)
- Commerce Development node (see Table 4-37 on page 144)

Table 4-29 Conceptual Web Server node

Subject	Conceptual node: Web Server
Description	A process that supports the use and management of Web pages and other Internet-related material.
General	The responsibilities of the Web server node include receiving and serving HTTP/HTTPS request from the browser client.
Users and Presentation	This node serves CEP customers and administrators. The presentation services are only browser based.
Performance and Capacity	The nonfunctional requirements and volumetric must be handled by this node. The disk capacity and caching configuration ensure reaching the transaction rate for the average/peak transaction rate according to the number of customers.
Availability	24x7.

Subject	Conceptual node: Web Server
Cost and Benefit	N/A.
Security	SSL software support. In case of performance problems, hardware encryption components can be added. The node access is defined by the CEP security policies. There is a well-defined relationship between Web Server nodes and App Server nodes.
Risk	It is mission critical since it serves all customer requests accessing the CEP site.
Node Management	Follows the CEP node management standards for remote management, recovery time, problem management, software distribution, and configuration management.

Table 4-30 Conceptual Application Server node

Subject	Conceptual node: App Server
Description	This node provides the infrastructure for application logic and may be part of a Web application server node. It is capable of running both presentation and business logic but generally does not serve HTTP requests. When used with a Web server redirector, the application server node will run both presentation and business logic. In other situations, it may be used for business logic only.
General	The responsibilities of the App Server include: receiving requests from the clients, selecting and executing the appropriate business logic based on these requests, coordinating with external services (for example, the LDAP directory) to retrieve data and execute external applications, and finally formulating the response and dispatching it back to the client. To meet these requirements, the application servers provide a range of dynamic page construction, business logic processing, data access, external application integration, session management, load balancing, and fail-over services.
Users and Presentation	The administration client provides a Java-based GUI to administer the App Server node.
Performance and Capacity	The nonfunctional requirements and volumetric must be handled by this node. The disk capacity and caching configuration ensure reaching the transaction rate for the average/peak transaction rate according to the number of customers.
Availability	24x7.

Subject	Conceptual node: App Server
Cost and Benefit	N/A.
Security	SSL is strictly enforced and maintenance access to the App Server runtime is granted based on the CEP security policy definitions. User access is authenticated against the Directory node.
Risk	It is mission critical and serves presentation and business logic of customer requests.
Node Management	Follows the CEP node management standards for remote management, recovery time, problem management, software distribution, and configuration management.

Table 4-31 Conceptual Payment node

Subject	Conceptual node: Payment
Description	This node provides services to process and manage payments in online, offline and batch mode. It is able to run presentation and business logic.
General	The responsibilities of the Payment node include receiving a request to process the payment details, approving the payment, managing deposits and handling a set of merchants including brands information. In addition, it provides online, offline, and batch mode capabilities for payment processing.
Users and Presentation	Browser-based payment application. Interfaces ensure that the base administration services can be integrated into e-business applications.
Performance and Capacity	The nonfunctional requirements and volumetric must be handled by this node. The disk capacity and caching configuration ensure reaching the transaction rate for the average/peak transaction rate according to the number of payment transactions.
Availability	24x7.
Cost and Benefit	Cost per SET transaction is about 50 cents. Additional transaction fees might be considered when accessing payment provider details.
Security	SSL is strictly enforced and maintenance access to the runtime environment is granted based on the CEP security policy definitions.
Risk	Mission critical.

Subject	Conceptual node: Payment
Node Management	Follows the CEP node management standards for remote management, recovery time, problem management, software distribution, and configuration management.

Table 4-32 Conceptual Directory node

Subject	Conceptual node: Directory
Description	The Directory node is a centralized, enterprise-wide directory structure optimized for the rapid retrieval of search results. It normally provides a separate multi-tier user administration function for updating the directory.
General	Lightweight Directory Access Protocol (LDAP) refers to the protocol that is used to communicate from a calling program (running on a node such as a Commerce Server) and a Directory node. Information is kept on the LDAP-based directory node about such topics as people and/or services. For example, the directory could be used to store information needed to identify registered shoppers (referred to as authentication). It could also be used to store information about what functions those shoppers are allowed to perform after being identified (referred to as authorization).
Users and Presentation	The administration client is browser based to configure the directory settings. The maintenance of LDAP entries is handled through a Java based GUI.
Performance and Capacity	The nonfunctional requirements and volumetric must be handled by this node. The disk capacity and caching configuration ensure reaching the transaction rate for the average/peak transaction rate according to the number of customers.
Availability	24x7.
Cost and Benefit	N/A
Security	SSL is strictly enforced and maintenance access to the directory server runtime is granted based on the CEP security policy definitions.
Risk	Mission critical.
Node Management	Follows the CEP node management standards for remote management, recovery time, problem management, software distribution, and configuration management.

Table 4-33 Conceptual Data Server node

Subject	Conceptual node: Data Server
Description	The Data Server node's function is to provide persistent data storage and retrieval in support of the user-to-business transactional interaction. The data stored is relevant to the specific business interaction, for example, bank balance, insurance information, current purchase by the user, etc.
General	It is important to note that the mode of database access is perhaps the most important factor determining the performance of this Web application, in all but the simplest cases. One approach is to collapse the database accesses into a single call or very few calls. This can be achieved via coding and invoking Java stored procedure calls on the database.
Users and Presentation	Administration facilities are Java-based GUIs. CEP database administrators have access to the production nodes.
Performance and Capacity	The nonfunctional requirements and volumetric must be handled by this node. The disk capacity and caching configuration ensure reaching the transaction rate for the average/peak transaction rate.
Availability	24x7.
Cost and Benefit	N/A.
Security	Access to the data server runtime is granted based on the CEP security policy definitions.
Risk	Mission critical.
Node Management	Follows the CEP node management standards for remote management, recovery time, problem management, software distribution, and configuration management.

Table 4-34 Conceptual Collaboration node

Subject	Conceptual node: Collaboration
Description	In e-business collaboration scenarios, the collaboration node provides a convenient place for supply chain members to collaborate on process improvements, troubleshooting, etc.

Subject	Conceptual node: Collaboration
General	On the demand side, it supports contract negotiations. Whatever the task at hand, it increases efficiency, makes it much easier for teams to collaborate across organizational boundaries, and enables the capture and reuse of best practices in dealing with suppliers, partners, and customers alike.
Users and Presentation	Browser-based administration services. Customers can access collaboration services using a browser.
Performance and Capacity	The nonfunctional requirements and volumetric must be handled by this node.
Availability	24x7.
Cost and Benefit	N/A
Security	SSL is strictly enforced in compliance with the CEP security policy.
Risk	Mission critical since business resellers are using the collaboration feature for negotiations.
Node Management	Follows the CEP node management standards for remote management, recovery time, problem management, software distribution, and configuration management.

Table 4-35 Conceptual Instant Messaging node

Subject	Conceptual node: Instant Messaging
Description	The Instant Messaging node provides real-time collaboration product for business, offering a customizable, standards-based platform for instant awareness, real-time communication (chat) and online meetings with application sharing and white-boarding capabilities.
General	Awareness is the focal point of the instant messaging node. First, it enables members of the online community to know who else is using the e-business collaboration environment - coworkers, partners, or customers. It then helps partners communicate in a variety of ways, from one-on-one instant messages to ad hoc virtual meetings. It also enables users to share live documents, applications, and a white board; for example, a seller can share a spreadsheet with a buyer, and the only application the buyer needs on her computer is a Java-aware Web browser.

Subject	Conceptual node: Instant Messaging
Users and Presentation	Browser-based administration services. Customers can access instant messaging services using a browser.
Performance and Capacity	The nonfunctional requirements and volumetric must be handled by this node.
Availability	08:00 - 17:00.
Cost and Benefit	CEP network bandwidth must handle the additional traffic and data volume for instant messaging streaming.
Security	SSL is strictly enforced. CEP extends the internal instant messaging capabilities, requiring further port and firewall settings.
Risk	Critical for customer support center.
Node Management	Follows the CEP node management standards for remote management, recovery time, problem management, software distribution, and configuration management.

Table 4-36 Conceptual Portal Development node

Subject	Conceptual node: Portal Development
Description	The Portal Development node provides the infrastructure and services to develop, test, and debug WebSphere Portal applications.
General	The responsibilities of the Portal Development node include maintaining source code for WebSphere Portal assets (HTML, JSP, code, portlets, data assets), testing and debugging applications, repository management, etc.
Users and Presentation	It serves WebSphere Portal developers and provides an integrated workbench for Java code and front-end design.
Performance and Capacity	Performance requirements demand a powerful developer workstation with at least 1 GB of memory. Disk capacity can handle the necessary software components and data volume.
Availability	08:00 - 19:00.
Cost and Benefit	N/A.
Security	None.
Risk	N/A.

Subject	Conceptual node: Portal Development
Node Management	Follows the CEP node management standards for remote management, recovery time, problem management, software distribution, and configuration management.

Table 4-37 Conceptual Commerce Development node

Subject	Conceptual node: Commerce Development
Description	The Commerce Development node provides the infrastructure and services to develop, test and debug WebSphere Commerce applications including Java code, JSPs and data assets.
General	The responsibilities of the Commerce Development node include maintaining source code for WebSphere Commerce assets (HTML, JSP, code, data assets), testing and debugging applications, repository management, etc.
Users and Presentation	It serves WebSphere Commerce developers and provides an integrated workbench for Java code and front-end design.
Performance and Capacity	Performance requirements demand a powerful developer workstation with at least 1 GB of memory. Disk capacity can handle the necessary software components and data volume.
Availability	08:00 - 19:00.
Cost and Benefit	N/A.
Security	Commerce development environment does not support SSL for testing purposes. Workstations cannot publish stores using WebSphere Commerce Studio V 5.4.
Risk	N/A
Node Management	Follows the CEP node management standards for remote management, recovery time, problem management, software distribution, and configuration management.

Specified model

Based on the given conceptual model, we refine the nodes by specifying the characteristics within the boundary of the associated domain. At that stage technological limitations are fully taken into account but the detailed choice of

technology is not made. The following factors affect the description of the specified nodes depending on various levels of abstraction:

- ▶ Detailed specifications of connections, computer system, operating systems characteristics, and nonfunctional characteristics, communications protocols, middleware, quantity, etc.
- ▶ Systems management style (centralized, distributed), and systems management protocol (for example, SNMP, CMIP).
- ▶ Software distribution method (for example, push, pull, attended, unattended, number of servers, etc.).
- ▶ Help desk, problem management, configuration management, change management, performance management, network management, and other system management procedures, etc.
- ▶ Simulations and prototypes are developed and run to verify design decisions, etc.

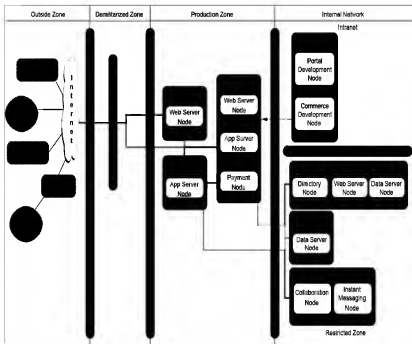


Figure 4-5 Specified model of the commerce enabled portal solution

Figure 4-5 on page 145 shows the nodes that are specified in the following tables. The naming conventions for the specified nodes are listed in Example 4-1.

Example 4-1 Naming convention for the specified operational model

1. **S** indicates it is a node of the specified operational model.
2. **P/R/D** indicates it belongs to the production zone, the development domain within the Intranet, or the restricted zone.
3. An abbreviation for the service, for example WCS stands for WebSphere Commerce.
4. An optional number to distinguish similar nodes within a CEP zone.

The node interaction matrix is shown in Table 4-38. It describes the relationship of specified nodes with their characteristics, for example, which type communication protocol is used.

Table 4-38 Node interaction matrix

From	To	Characteristics
User	DMZ	HTTP (80), HTTPS (443)
DMZ	SPHTTP01	HTTP (80), HTTPS (443)
DMZ	SPHTTP02	HTTP (80), HTTPS (443)
Admin	SRHTTP01	HTTP(80), HTTPS(443)
SPHTTP01	SPWPS	HTTP (80), HTTPS (443)
SPWPS	SPHTTP02	HTTP (80), HTTPS (443)
SPWPS	SRDO	HTTPS(443)
SPWPS	SRST	HTTPS(443), Native (1533)
SPWPS	SRLDAP	LDAP(389), LDAPS(636)
SPWPS	SRDB02	DB2 client/server connection port 50000
SPHTTP02	SPWCS	HTTP (80), HTTPS (443)
SPWCS	SPPAY	
SPWCS	SRLDAP	LDAP(389), LDAPS(636)

From	To	Characteristics
SPWCS	SRDO	HTTP(80), HTTPS(443), Native
SPWCS	SRST	HTTP(80), HTTPS(443), Native
SPWCS	SRDB02	DB2 client/server connection port 50000
SPPAY	SRDB02	DB2 client/server connection port 50000
SPHTTP02	SPPAY	HTTP(80), HTTPS(443)
SRDO	SRLDAP	LDAP(389), LDAPS(636)
SRST	SRLDAP	LDAP(389), LDAPS(636)
SRHTTP01	SRLDAP	
SRLDAP	SRDB01	Local DB

The following tables describe the specified nodes in detail.

Table 4-39 Specified SPHTTP01 node

Subject	Specified node: SPHTTP01 (Web Server)
Description	This node serves portal requests of the CEP site. 24x7
Presentation Function	The administration services are browser based leveraging HTML and Java applets.
Processing Function	Runs five or more HTTP services to serve HTTP requests. Delivers HTTP response streams to the client browser.
Data	Logs HTTP requests on the local file system. Serves up to 200 concurrent users per hour.
Infrastructure	Ethernet access with at least 100 Mbps LAN.
Presentation Services	None.
Processing Services	IBM HTTP Web server required to handle HTTP requests. 128-bit SSL encryption.
Data Services	Handles <= 1000 connections.

Subject	Specified node: SPHTTP01 (Web Server)
Hardware	CEP productive server profile. For scalability reasons the hardware must support up to two processors.
Operating System	Supports <= 512 threads. Multitasking multi-processor operating system.
Connections	Based on TCP/IP serving HTTP and HTTPS requests. Access to file system using native operating services.
Management	Tivoli systems management software 4.x or higher.

Table 4-40 Specified SPHTTP02 node

Subject	Specified node: SPHTTP02 (Web Server)
Description	This node serves WebSphere Commerce requests of CEP only.
Presentation Function	The administration services are browser based leveraging HTML and Java applets.
Processing Function	Runs 50 or more HTTP services to serve HTTP requests. Delivers HTTP response streams to the client browser.
Data	Logs HTTP requests on the local file system. Serves up to 300 concurrent users per hour. 24x7.
Infrastructure	Ethernet access with at least 100 Mbps LAN.
Presentation Services	None.
Processing Services	IBM HTTP Web server required to handle HTTP requests. 128-bit SSL encryption.
Data Services	Handles <= 2000 connections.
Hardware	CEP productive server profile. For scalability reasons, the hardware must support up to two processors.
Operating System	Supports <= 512 threads. Multitasking multi-processor operating system.
Connections	Based on TCP/IP serving HTTP and HTTPS requests. Access to file system using native operating services.
Management	Tivoli systems management software 4.x or higher.

Table 4-41 Specified SRHTTP01 node

Subject	Specified node: SRHTTP01 (Web Server)
Description	This node serves directory administration requests only. 08:00 - 18:00
Presentation Function	The administration services are browser based leveraging HTML and Java applets.
Processing Function	Runs two or more HTTP services to serve HTTP requests. Delivers HTTP response streams to the client browser.
Data	Logs HTTP requests on the local file system.
Infrastructure	Ethernet access with at least 100 Mbps LAN.
Presentation Services	None.
Processing Services	IBM HTTP Web server required to handle HTTP requests. 128-bit SSL encryption.
Data Services	N/A.
Hardware	CEP productive server profile.
Operating System	Multitasking multi-processor operating system.
Connections	Based on TCP/IP serving HTTP and HTTPS requests. Access to file system using native operating services.
Management	Tivoli systems management software 4.x or higher.

Table 4-42 Specified SPWPS node

Subject	Specified node: SPWPS (App Server)
Description	This node serves portal requests and renders the client page. It supports multi-channel access serving HTML and WAP WML clients. 24x7.
Presentation Function	Browser-based configuration and administration. Application server services are managed through a Java Client.
Processing Function	Processes HTTP request from an HTML or WAP WML capable client browser. Multi-threading services to share the load of serving customer requests. Based on J2EE open standards including Web services.

Subject	Specified node: SPWPS (App Server)
Data	Logs application server relevant data on the local file system. Uses database repository to maintain application server configuration.
Infrastructure	Ethernet access with at least 100 Mbps LAN.
Presentation Services	Supports JavaServer Pages, WML and HTML.
Processing Services	Rendering engine to generate markup language for HTML and WML. J2EE compliant runtime environment leveraging Java services. WebSphere Portal Enable/Extend
Data Services	JDBC driver to access database relevant information. Handles data as defined by the nonfunctional requirements.
Hardware	CEP productive server profile.
Operating System	Multitasking multi-processor operating system.
Connections	Based on TCP/IP serving HTTP and HTTPS requests. Access to file system using Java services. JDBC driver from database vendor.
Management	Tivoli systems management software 4.x or higher.

Table 4-43 Specified SPWCS node

Subject	Specified node: SPWCS (App Server)
Description	This node serves commerce requests and renders the client page. It supports multi-channel access serving HTML and WAP WML clients. 24x7.
Presentation Function	Browser-based configuration and administration. Commerce instance data is managed through a Java Client.
Processing Function	Processes HTTP request from a HTML or WAP WML capable client browser. Multi-threading services to share the load of serving customer requests. Based on J2EE open standards including Web services.

Subject	Specified node: SPWCS (App Server)
Data	Logs application server relevant data on the local file system. Uses database repository to maintain application server configuration. Store information and data assets are kept in the commerce database.
Infrastructure	Ethernet access with at least 100 Mbps LAN.
Presentation Services	Supports JavaServer Pages, WML and HTML.
Processing Services	Generates markup language for HTML and WML. J2EE compliant runtime environment leveraging Java services. WebSphere Commerce Server Business Edition.
Data Services	JDBC driver to access database relevant information. Handles data as defined by the nonfunctional requirements.
Hardware	CEP productive server profile.
Operating System	Multitasking multi-processor operating system.
Connections	Based on TCP/IP serving HTTP and HTTPS requests. Access to file system using Java services. JDBC driver from database vendor.
Management	Tivoli systems management software 4.x or higher.

Table 4-44 Specified SPPAY node

Subject	Specified node: SPPAY (App Server)
Description	This node serves payment requests and renders payment related administration pages. It supports HTML clients. 24x7.
Presentation Function	Browser-based configuration and administration.
Processing Function	Processes HTTP request from a HTML capable client browser. Based on J2EE open standards.
Data	Logs application server relevant data on the local file system. Uses database repository to maintain payment server configurations and payment processing information.
Infrastructure	Ethernet access with at least 100 Mbps LAN.

Subject	Specified node: SPPAY (App Server)
Presentation Services	Supports JavaServer Pages and HTML.
Processing Services	Generates markup language for HTML. J2EE compliant runtime environment leveraging Java services. WebSphere Payment Manager.
Data Services	JDBC driver to access database relevant information. Handles data as defined by the nonfunctional requirements.
Hardware	CEP productive server profile.
Operating System	Multitasking multi-processor operating system.
Connections	Based on TCP/IP serving HTTP and HTTPS requests. Access to file system using Java services. JDBC driver from database vendor.
Management	Tivoli systems management software 4.x or higher.

Table 4-45 Specified SRLDAP node

Subject	Specified node: SRLDAP (Directory)
Description	This node serves the enterprise directory of CEP. 24x7.
Presentation Function	Browser-based configuration and administration. Directory entities are managed through a Java Client.
Processing Function	Processes LDAP and LDAP requests. Multi-threading services to share the load of serving directory requests. Based on J2EE and open standards.
Data	Logs directory relevant data on the local file system. Uses database repository to maintain directory configuration.
Infrastructure	Ethernet access with at least 100 Mbps LAN
Presentation Services	Supports HTML for administration purposes.
Processing Services	Generates markup language for HTML. J2EE compliant runtime environment leveraging Java services. Serves LDAP and LDAPS V3.0 requests. IBM Directory Server
Data Services	JDBC driver to access database relevant information. Handles data as defined by the nonfunctional requirements.

Subject	Specified node: SRLDAP (Directory)
Hardware	CEP productive server profile.
Operating System	Multitasking multi-processor operating system.
Connections	Based on TCP/IP serving HTTP, HTTPS, LDAP, and LDAPS requests. Access to file system using Java services. JDBC driver from database vendor.
Management	Tivoli systems management software 4.x or higher.

Table 4-46 Specified SRDB01 node

Subject	Specified node: SRDB01 (Data Server)
Description	This node is the data store for the enterprise directory of CEP which is optimized for read access. 24x7.
Presentation Function	None
Processing Function	Is able to serve 10000 directory database requests per hour. Database provides performance optimizations to handle the load.
Data	Use of native relational database technology.
Infrastructure	Ethernet access with at least 100 Mbps LAN. Optimized hard disk access.
Presentation Services	None
Processing Services	Relational database supporting SQL. DB2
Data Services	JDBC support. Database is able to handle 80 concurrent connections and provides stored procedures. Handles data as defined by the nonfunctional requirements
Hardware	CEP productive server profile.
Operating System	Multitasking multi-processor operating system
Connections	Based on TCP/IP. Native database driver from database vendor including JDBC support
Management	Tivoli systems management software 4.x or higher.

Table 4-47 Specified SRDB02 node

Subject	Specified node: SRDB02 (Data Server)
Description	This node is the data store for all e-business applications within CEP. 24x7.
Presentation Function	None
Processing Function	Is able to serve CEP database requests per hour. Database provides performance optimizations to handle the load.
Data	Use of native relational database technology.
Infrastructure	Ethernet access with at least 100 Mbps LAN. Optimized hard disk access.
Presentation Services	None
Processing Services	Relational database supporting SQL. DB2
Data Services	JDBC support including 800 concurrent connections. Database provides Stored Procedures. Handles data as defined by the nonfunctional requirements.
Hardware	CEP productive server profile.
Operating System	Multitasking multi-processor operating system.
Connections	Based on TCP/IP. Native database driver from database vendor including JDBC support.
Management	Tivoli systems management software 4.x or higher.

Table 4-48 Specified SRDO node

Subject	Specified node: SRDO (Collaboration)
Description	This node serves collaboration requests of CEP. 24x7.
Presentation Function	Lotus Quickplace base services.
Processing Function	Lotus Quickplace base services.
Data	Lotus Quickplace base data.
Infrastructure	Ethernet access with at least 100 Mbps LAN.

Subject	Specified node: SRDO (Collaboration)
Presentation Services	see Lotus Quickplace.
Processing Services	Lotus Quickplace base services.
Data Services	Requires Lotus Domino Notes®.
Hardware	CEP productive server profile.
Operating System	Multitasking multi-processor operating system.
Connections	Based on TCP/IP. Lotus Domino Notes specific extensions.
Management	Tivoli systems management software 4.x or higher.

Table 4-49 Specified SRST node

Subject	Specified node: SRST (Instant Messaging)
Description	This node serves instant messaging requests of CEP. 08:00 - 18:00
Presentation Function	Lotus Sametime base services.
Processing Function	Lotus Sametime base services.
Data	Lotus Sametime base data.
Infrastructure	Ethernet access with at least 100 Mbps LAN.
Presentation Services	See Lotus Sametime.
Processing Services	Lotus Sametime base services.
Data Services	Requires Lotus Domino Notes.
Hardware	CEP productive server profile.
Operating System	Multitasking multi-processor operating system.
Connections	Based on TCP/IP. Lotus Domino Notes specific extensions.
Management	Tivoli systems management software 4.x or higher.

Table 4-50 Specified SDWPS node

Subject	Specified node: SDWPS (Portal Development)
Description	This node serves the requirements to develop portal applications. 08:00 - 18:00

Subject	Specified node: SDWPS (Portal Development)
Presentation Function	See WebSphere Studio Application Developer.
Processing Function	See WebSphere Studio Application Developer. See WebSphere Portal Enable/Extend.
Data	Use of native relational database technology.
Infrastructure	Ethernet access with at least 100 Mbps LAN. Optimized hard disk access.
Presentation Services	See WebSphere Studio Application Developer.
Processing Services	WebSphere Studio Application Developer WebSphere Portal Enable/Extend DB2
Data Services	JDBC support See WebSphere Studio Application Developer.
Hardware	CEP developer workstation profile.
Operating System	Multitasking multi-processor operating system.
Connections	Based on TCP/IP. Native database driver from database vendor including JDBC support.
Management	Tivoli systems management software 4.x or higher.

Table 4-51 Specified SDWPS node

Subject	Specified node: SDWCS (Commerce Development)
Description	This node serves the requirements for developing commerce applications. 08:00 - 18:00
Presentation Function	See WebSphere Commerce Studio.
Processing Function	See WebSphere Commerce Studio.
Data	Use of native relational database technology.
Infrastructure	Ethernet access with at least 100 Mbps LAN. Optimized hard disk access.
Presentation Services	See WebSphere Commerce Studio.
Processing Services	WebSphere Studio Application Developer V5. DB2

Subject	Specified node: SDWCS (Commerce Development)
Data Services	JDBC support. See WebSphere Commerce Studio.
Hardware	CEP developer workstation profile.
Operating System	Multitasking multi-processor operating system.
Connections	Based on TCP/IP. Native database driver from database vendor including JDBC support.
Management	Tivoli systems management software 4.x or higher.

Physical model

Based on the given specified model, we refine the physical nodes by specifying the characteristics within the boundary of the associated domain. At that stage technological limitations are fully taken into account and the detailed choice of technology is made. The following factors affect the description of the physical nodes depending on various levels of abstraction:

- ▶ A UNIX operating system becomes IBM-AIX, SUN-OS, RISC processor, a 10 Mbps link, etc.
- ▶ Systems management tools (for example, Tivoli products, IBM NetView®, HP OpenView, etc.).
- ▶ Nonfunctional characteristics are specified, for example latency of network connections, availability, performance, network redundancy routes, and equipment.
- ▶ High availability systems, for example IBM HACMP, multi-protocol router Cisco 7000 (10 port) TCP/IP, SNA, IPX, OSPF, APPN-capable router

The physical operational model provides the necessary information to set up the runtime environment and development environment. Figure 4-6 on page 158 sketches the physical operational model of the CEP site.

In summary, eight nodes have been chosen for the CEP commerce enabled portal solution. The specified nodes have been placed onto the appropriate physical nodes to fulfill the functional and nonfunctional requirements.

Note: The sizing of the physical nodes is fictitious and does not reflect the sizing requirements in a real-life production scenario.

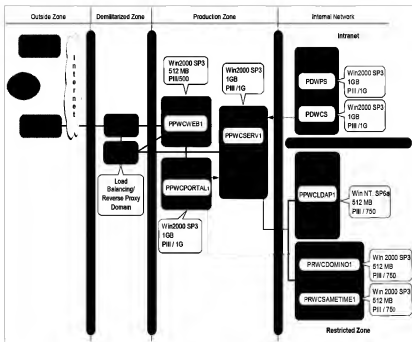


Figure 4-6 Physical model of the commerce enabled portal solution

The naming conventions for the physical nodes are listed in Example 4-2.

Example 4-2 Naming convention for the physical operational model

1. P indicates it is a node of the physical operational model.
2. P/R/D, indicates it belongs to the production zone, the development domain within the Intranet, or the restricted zone.
3. An abbreviation for the machines host name, for example WCSERV1 stands for WebSphere Commerce.
4. An optional number to distinguish similar nodes within a CEP zone.

The node interaction matrix is shown in Table 4-52 on page 159. It describes the relationship of physical nodes with their characteristics, for example, which type communication protocol is used.

Table 4-52 Node interaction matrix

From	To	Characteristics
User	DMZ	HTTP, HTTPS
DMZ	PPWCWEB1	HTTP, HTTPS
DMZ	PPWCSESV1	HTTP,HTTPS
PPWCWEB1	PPWCPORTAL1	HTTP,HTTPS
PPWCPORTAL1	PPWCSESV1	HTTP,HTTPS
PPWCPORTAL1	PPWCLDAP1	LDAPS, DB2
PPWCSESV1	PPWCLDAP1	LDAPS,DB2
PPWCPORTAL1	PRWCDOMINO1	HTTP,HTTPS,SMTP
PPWCPORTAL1	PRWCSAMETIME1	
PPWCSESV1	PRWCDOMINO1	HTTP,HTTPS,SMTP
PPWCSESV1	PRWCDOMINO1	

The following tables describe the specified nodes in detail.

Table 4-53 Physical node PPWCWEB1

Subject	Physical node: PPWCWEB1
Specified nodes	SPHTTP01
Data	Images and static HTML pages for the CEP portal.
Infrastructure	None
Software Packages	IBM HTTP Server 1.3.19.4 IBM SSL 128-bit encryption Tivoli System Management 4.x
Data Services	None
Hardware	500 MHz Pentium 6 GB disk 512 MB memory Ethernet Card
Operating System	Windows 2000 with Service Pack 3
Connections	Ethernet 10BaseT, TCP/IP only
Management	Remote system management via SNMP

Table 4-54 Physical node PPWCSESV1

Subject	Physical node: PPWCSESV1
Specified nodes	SPHTTP02, SPWCS, SPPAY
Data	HTML pages, images (GIF, JPEG), PDF files, WebSphere Commerce data assets.
Infrastructure	None
Software Packages	IBM HTTP Server 1.3.19.4 IBM SSL 128-bit encryption WebSphere Commerce Server Business Edition V5.4 WebSphere Commerce Server FixPak V 5.4.0.5 Commerce Enhancement Pack April 2003 WebSphere Payment Manager 3.1.3.1 WebSphere Application Server Advanced Edition 4.0.5 DB2 V7.2 EE and FixPak 7 Tivoli System Management 4.x
Data Services	DB2 JDBC V2.0
Hardware	1 GHz Pentium 6 GB disk 1 GB memory Ethernet Card
Operating System	Windows 2000 with Service Pack 3
Connections	Ethernet 10BaseT, TCP/IP only
Management	Remote system management via SNMP

Table 4-55 Physical node PPWCPORTAL1

Subject	Physical node: PPWCPORTAL1
Specified nodes	SPWPS
Data	WebSphere Portal data assets
Infrastructure	None
Software Packages	WebSphere Portal V4.2.1 Enable WebSphere Application Server Advanced Edition 4.0.5 DB2 V7.2 EE Client and FixPak 7 Tivoli System Management 4.x
Data Services	DB2 JDBC V2.0

Subject	Physical node: PPWCPORTAL1
Hardware	1 GHz Pentium 6 GB disk 1 GB memory Ethernet Card
Operating System	Windows 2000 with Service Pack 3
Connections	Ethernet 10BaseT, TCP/IP only
Management	Remote system management via SNMP

Table 4-56 Physical node PPWCLDAP1

Subject	Physical node: PPWCLDAP1
Specified nodes	SRDB01, SRDB02, SRHTTP01, SRLDAP
Data	IBM Directory Server data assets CEP database assets
Infrastructure	None
Software Packages	IBM HTTP Server 1.3.19.4 IBM SSL 128-bit encryption IBM Directory Server V4.1 DB2 V 7.2 EE and FixPak 7 Tivoli System Management 4.x
Data Services	DB2 JDBC 2.0
Hardware	750 MHz Pentium 6 GB disk 512 MB memory Ethernet Card
Operating System	Windows NT® with Service Pack 6a
Connections	Ethernet 10BaseT, TCP/IP only
Management	Remote system management via SNMP

Table 4-57 Physical node PPWCSAMETIME1

Subject	Physical node: PPWCSAMETIME1
Specified node	SRST
Data	Lotus Sametime data assets
Infrastructure	Firewall port assignments for instant messaging and streaming (1533)

Subject	Physical node: PPWCSAMETIME1
Software Packages	Lotus Sametime V 2.5 Lotus Domino 5.0.10 Tivoli System Management 4.x
Data Services	Lotus Domino Notes
Hardware	750 MHz Pentium 6 GB disk 512 MB memory Ethernet Card
Operating System	Windows 2000 with Service Pack 3
Connections	Ethernet 10BaseT, TCP/IP only
Management	Remote system management via SNMP

Table 4-58 Physical node PPWCDDOMINO1

Subject	Physical node: PPWCDDOMINO1
Specified node	SRDO
Data	Lotus Domino data assets
Infrastructure	None
Software Packages	Lotus Quickplace V 2.0.8 Lotus Domino V 5.0.11 Tivoli System Management 4.x
Data Services	Lotus Domino
Hardware	750 MHz Pentium 6 GB disk 512 MB memory Ethernet Card
Operating System	Windows 2000 with Service Pack 3
Connections	Ethernet 10BaseT, TCP/IP only
Management	Remote system management via SNMP

Table 4-59 Physical node PDWPS

Subject	Physical node: PDWPS
Specified node	SDWPS

Subject	Physical node: PDWPS
Data	WebSphere Studio Application Developer and WebSphere Portal data assets
Infrastructure	Configuration of port 8080 to access development environment
Software Packages	WebSphere Studio Application Developer 5.0.2 (PTF 001) WebSphere Portal V4.2.1 Enable WebSphere Portal Toolkit V4.2.5 DB2 V 7.2 EE and FixPak 7 Tivoli System Management 4.x
Data Services	DB2 JDBC 2.0
Hardware	1.8 GHz Pentium 6 GB disk 2 GB memory Ethernet Card
Operating System	Windows 2000 with Service Pack 3
Connections	Ethernet 10BaseT, TCP/IP only
Management	Remote system management via SNMP.

Table 4-60 Physical node PDWCS

Subject	Physical node: PDWCS
Specified node	SDWCS
Data	WebSphere Commerce Studio data assets
Infrastructure	Configuration of port 8080 to access development environment
Software Packages	WebSphere Commerce Studio Business Edition V5.4 WebSphere Commerce Server FixPak V 5.4.0.5 Commerce Enhancement Pack April 2003 DB2 V 7.2 EE and FixPak 7 Tivoli System Management 4.x
Data Services	DB2 JDBC 2.0
Hardware	1 GHz Pentium 6 GB disk 1 GB memory Ethernet Card

Subject	Physical node: PDWCS
Operating System	Windows 2000 with Service Pack 3
Connections	Ethernet 10BaseT, TCP/IP only
Management	Remote system management via SNMP



Implement the runtime environment

This chapter describes how to implement a multi-tiered WebSphere Commerce V5.4 and WebSphere Portal V4.2.1 runtime environment used to deploy the ITSO B2B CEP store working example. The procedures documented include the WebSphere Commerce FixPak V5.4.0.5, IBM Commerce Enhancement Pack - April 2003 Edition, and many other FixPak levels beyond the WebSphere Commerce generally available release, workarounds and best practices.

This chapter is organized into the following sections:

- ▶ Planning for implementation and deployment
- ▶ WebSphere Commerce node implementation
- ▶ Directory Server (LDAP) node implementation
- ▶ WebSphere Portal node implementation
- ▶ Enabling single sign-on between WebSphere Portal and WebSphere Commerce
- ▶ Deploying the ITSO B2B CEP store
- ▶ DB2 Server node implementation
- ▶ Remote Web server node implementation

5.1 Planning for implementation and deployment

It is important to prepare in advance of the implementation of the runtime environment. For example, you will need to obtain static TCP/IP addresses and host names in advance, since these are not easily changed once everything is configured.

This section describes the key considerations for planning for implementation, such as the network configuration, hardware, and software used in the runtime environment, and the high-level process we used to implement the runtime environment. The runtime environment implementation is part of the deployment process.

This section includes the following topics:

- ▶ Network environment
- ▶ Hardware used within the ITSO test environment
- ▶ Software used within the ITSO test environment
- ▶ VMWare or Ghost
- ▶ High-level implementation procedure
- ▶ Downloading the IBM Commerce Enhancement Pack - April 2003 Edition

5.1.1 Network environment

The ITSO runtime environment is depicted in Figure 5-1 on page 167 with a focus on the numbered nodes. The nodes displayed in Figure 5-1 on page 167 and listed in Table 5-1 are installed on the Windows 2000 Server platform. The focus of this redbook is on integration of WebSphere Commerce and WebSphere Portal.

Table 5-1 Nodes and host names for the ITSO working example runtime environment

Diagram #	Node name	Host name
1	WebSphere Commerce node	wcserv1
2	Directory Server node	wldap1
3	WebSphere Portal node	wcportal1
4	DB2 Server node	wcdb2
5	Web Server node	wcweb1

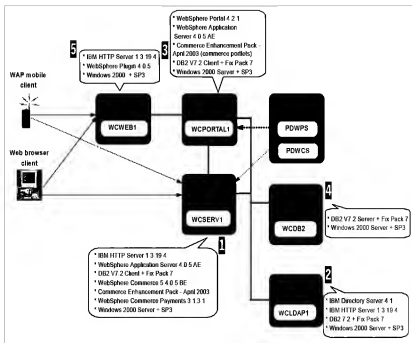


Figure 5-1 ITSO runtime environment and nodes implemented for the working example

5.1.2 Hardware used within the ITSO test environment

We recommend that you refer to the product documentation for the official guidelines and prerequisite hardware requirements. The hardware we used within the ITSO runtime environment for the stated nodes is listed in the following sections.

WebSphere Commerce node

We used the following hardware for the WebSphere Commerce node:

- ▶ IBM @server xSeries® 230 (8658-61Y)
 - 1 CPU, 1 GHz Intel PIII
 - 1 GB RAM
 - 18 GB hard disk
 - 1 IBM Ethernet (NetFinity Fault Tolerance PCI Adapter)

2 Directory Server node

We used the following hardware for the Directory Server node:

- ▶ IBM NetFinity 3000 (8476-41U)
 - 1 CPU, 450 MHz Intel PIII
 - 512 MB RAM
 - 4 GB hard disk
 - 1 IBM Ethernet Adapter

3 WebSphere Portal node

We used the following hardware for the WebSphere Portal node:

- ▶ IBM *@server*™ xSeries 230 (8658-61Y)
 - 1 CPU, 1 GHz Intel PIII
 - 1 GB RAM
 - 18 GB hard disk
 - 1 IBM Ethernet (NetFinity Fault Tolerance PCI Adapter)

4 DB2 Server node

We used the following hardware for the DB2 Server node:

- ▶ IBM *@server* xSeries 230 (8658-61Y)
 - 1 CPU, 1 GHz Intel PIII
 - 1 GB RAM
 - 18 GB hard disk
 - 1 IBM Ethernet (NetFinity Fault Tolerance PCI Adapter)

5 Web Server node

We used the following hardware for the Web Server node for the WebSphere Portal:

- ▶ IBM NetFinity 3000 (8476-41U)
 - 1 CPU, 450 MHz Intel PIII
 - 512 MB RAM
 - 4 GB hard disk
 - 1 IBM Ethernet Adapter

5.1.3 Software used within the ITSO test environment

Within the ITSO runtime environment, we used FixPak levels beyond the generally available release level of the required software components. We used the FixPak levels to address software issues and used the latest service levels available at the time of writing this redbook.

1 WebSphere Commerce node

Table 5-2 lists the software levels used on the WebSphere Commerce node within the ITSO runtime environment.

Table 5-2 Software levels used for the WebSphere Commerce node

Software	Original Level	FixPak level
Microsoft Windows 2000 Server	na	Service Pack 3
IBM DB2 UDB V7.2, Enterprise Edition	7.1.0.55 FixPak 5	7.1.0.68 FixPak 7
IBM HTTP Server	1.3.19	1.3.19.4 (WebSphere Application Server FixPak 5)
IBM WebSphere Application Server, Advanced Edition	4.0.1	4.0.5 (FixPak 5 and interim fixes)
IBM WebSphere Commerce V5.4, Business Edition	5.4	5.4.0.5 (FixPak 5.4.0.5)
IBM WebSphere Payment Manager (renamed to IBM WebSphere Commerce Payments)	3.1.2	3.1.3.1
IBM Commerce Enhancement Pack	April 2003 Edition	na

2 Directory Server node

Table 5-3 lists the software levels used on the Directory Server node within the ITSO runtime environment.

Table 5-3 Software levels used for the Directory Server node

Software	Original Level	FixPak level
Microsoft Windows 2000 Server	na	Service Pack 3
IBM DB2 UDB V7.2, Enterprise Edition	7.1.0.55 FixPak 5	7.1.0.68 FixPak 7
IBM HTTP Server	1.3.19	1.3.19.4 (WebSphere Application Server FixPak 5)
IBM Directory Server	4.1	na

3 WebSphere Portal node

Table 5-4 lists the software levels used on the WebSphere Portal node within the ITSO runtime environment.

Table 5-4 Software levels used for this redbook on the WebSphere Portal node

Software	Original Level	FixPak level
Microsoft Windows 2000 Server	na	Service Pack 3
IBM DB2 UDB V7.2, Client only	7.1.0.68 FixPak 7	na
IBM HTTP Server Note: This Web server will be installed on a separate node. We installed this on this node for ease of configuration of the httpd.conf. The server will then be disabled.	1.3.19	1.3.19.4 (WebSphere Application Server FixPak 5)
IBM WebSphere Application Server, Advanced Edition	4.0.1	4.0.5 (FixPak 4 and interim fixes)
IBM WebSphere Personalization included with WebSphere Portal Content Publisher	4.0.2	na
IBM WebSphere Portal Enable	4.2.1	na
IBM Commerce Enhancement Pack (commerce portlets)	April 2003 Edition	na

4 DB2 Server node

Table 5-5 lists the software levels used on the DB2 Server node within the ITSO runtime environment.

Table 5-5 Software levels used on the DB2 Server node

Software	Original Level	FixPak level
Microsoft Windows 2000 Server	na	Service Pack 3
IBM DB2 UDB V7.2, Enterprise Edition	7.1.0.55 FixPak 5	7.1.0.68 FixPak 7

5 Web Server node

Table 5-6 on page 171 lists the software levels used on the remote Web Server node for the WebSphere Portal within the ITSO runtime environment.

Table 5-6 Software levels used for the Web Server node

Software	Original Level	FixPak level
Microsoft Windows 2000 Server	na	Service Pack 3
IBM HTTP Server	1.3.19.1	1.3.19.4 (WebSphere Application Server FixPak 5)
IBM WebSphere Application Server, Advanced Edition, plug-in	4.0.1	4.0.5 (FixPak 5)

5.1.4 VMWare or Ghost

While developing the implementation procedure, we found it very useful to use VMWare 3.2 and Ghost 6.5 to take a snapshot of the installation by creating an image of the system. There are other utilities such as this on the market. Each utility has its advantages.

A VMWare system image is very nice in that it is portable to different systems. You can store multiple versions of virtual machines on the same system and easily start them (limited by disk space). When using VMWare, you do sacrifice a bit of performance. We used VMWare 3.2 and the Microsoft Sysprep utility to change the Windows SID. We found VMWare to be especially useful and amazingly compatible and reliable. For more information on VMWare, refer to the following URL:

<http://www.vmware.com/>

During the installation of the nodes in the ITSO runtime environment, we created zip files (backups) for various stages of the deployment.

Ghost allows for the imaging of systems, but is much more limited in moving images to a system of different hardware specs (device drivers). The big advantage of Ghost is that when you are done, you are running the native operating system on the hardware and can achieve better performance. We used the Ghost multicast feature to load systems of the network and created Microsoft DOS-based network client boot diskettes to connect to a Windows 2000 share where the Ghost image was stored. This can be used to capture (dump) or load from an image. For more information on Symantec Ghost, refer to the following URL:

<http://www.ghost.com/>

Using these utilities allowed us to go back to a previous state during the installation (provided an image was captured of the system). This can save a

tremendous amount of time and allow you to verify your knowledge of the environment before proceeding to deploy in a production environment.

5.1.5 High-level implementation procedure

The high-level runtime environment implementation procedure is as follows:

1. WebSphere Commerce node implementation
2. Directory Server (LDAP) node implementation
3. WebSphere Portal node implementation
4. Enabling single sign-on between WebSphere Portal and WebSphere Commerce
5. Deploying the ITSO B2B CEP store
6. DB2 Server node implementation
7. Remote Web server node implementation

We chose the order listed to avoid problems with software dependencies. Also, we wanted the procedure to be flexible for users who want to install the components on the same node. In some cases, the installation procedures run more smoothly when following the order listed. For example, the *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition* documents an environment where the database and application server are on the same node.

Once the WebSphere Commerce and WebSphere Portal nodes are working, we will back up and restore databases (WebSphere Application Server, WebSphere Commerce, WebSphere Commerce Payments, and WebSphere Portal) to the database server and configure the DB2 client/server connection.

5.1.6 Downloading the IBM Commerce Enhancement Pack - April 2003 Edition

We will use the IBM Commerce Enhancement Pack - April 2003 Edition throughout the implementation of the WebSphere Commerce, WebSphere Portal and LDAP nodes. This section describes packaging of the IBM Commerce Enhancement Pack - April 2003 Edition.

1. Download the Windows version of the IBM Commerce Enhancement Pack - April 2003 Edition to a temporary folder from the following URL:

<http://www.ibm.com/software/webservers/commerce/epacks/v54/>

You will be required to register.

There are two zip files you will need to download:

- WIN_April2003_CommerceEnhancementPack.zip

This zip contains the CommerceEnhancementPack_WIN_OCTOBER2002.zip and the WC_3D_Secure.zip.

- WIN_04-28-2003_cumulative_CommerceEnhancementPack.zip

This zip file contains the changes beyond the CommerceEnhancementPack_WIN_OCTOBER2002.zip.

2. Download the service level of the commerce enabled portal to be installed on top of the IBM Commerce Enhancement Pack - April 2003 Edition cumulative fix.

- a. Enter the following URL and refer to APAR JR18068.

http://www-1.ibm.com/support/docview.wss?rs=494&context=SSZLC2&q=4.2&uid=swg21107559&loc=en_US&cs=utf-8&lang=en

- b. Download WIN_JR18068.zip and the Integration Guide PDF from the bottom of the page.

3. Unzip the WebSphere CommerceEnabledPortal.zip to the c:\temp\cep directory. This zip is found in the CommerceEnhancementPack_WIN_OCTOBER2002.zip, which is zipped in the WIN_04-28-2003_cumulative_CommerceEnhancementPack.zip.

Refer to 5.2.15, "Commerce Enabled Portal - APAR JR18068" on page 198 for information on applying APAR files.

5.2 WebSphere Commerce node implementation

This section describes the high-level steps to install the WebSphere Commerce node within the ITSO test environment. Table 5-2 on page 169 lists the software levels used on the WebSphere Commerce node within the ITSO runtime environment.

The procedure outlined includes best practices and FixPaks beyond the original release of WebSphere Commerce V5.4.

Note: More detailed information can be found in the redbook *WebSphere Commerce V5.4 Handbook*, SG24-6567, or the product installation guides.

The high-level installation and configuration procedures for the WebSphere Commerce node are in the following order:

1. Windows 2000 Server installation
2. DB2 Server installation
3. IBM HTTP Server installation
4. WebSphere Application Server installation
5. WebSphere Commerce installation
6. WebSphere Commerce Payments installation
7. Commerce Enhancement Pack installation
8. WebSphere Commerce instance creation
9. Commerce Enhancement Pack post-install configuration
10. WebSphere Commerce administration tools verification
11. WebSphere Commerce Payments configuration
12. Database backup
13. Enable WebSphere Commerce portal adapter
14. Disable the WebSphere Commerce cache
15. Commerce Enabled Portal - APAR JR18068

5.2.1 Windows 2000 Server installation

In preparation for the installation of WebSphere Commerce and supporting components, ensure the following tasks have been completed:

1. Install Windows 2000 Server and Windows 2000 Service Pack 3.
2. Ensure an administrator user is logged in for installation of the WebSphere Commerce and supporting software with the following user rights:
 - Act as part of the operating
 - Create a token object
 - Increase quotas
 - Log on as a service
 - Replace a process level token
3. Install Internet Explorer 5.5 and service pack or higher.
4. Verify the configuration of the TCP/IP network (host name, IP address).

5.2.2 DB2 Server installation

This section highlights the key steps in installing and configuring the DB2 Server for use with WebSphere Application Server, WebSphere Commerce, and WebSphere Commerce Payments. In our example, we will configure the Commerce Application Server node first with a local database server.

The DB2 Server installation is organized as follows:

1. Install DB2 UDB V7.2 Enterprise Edition
2. Install DB2 V7 FixPak 7 (7.1.0.68)
3. Update JDBC level to JDBC2

Note: For detailed installation instructions, refer to the following:

- ▶ *WebSphere Commerce V5.4 Handbook, SG24-6567*
- ▶ *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Windows product guide*

Install DB2 UDB V7.2 Enterprise Edition

The high-level steps to install the IBM DB2 UDB V7.2, Enterprise Edition, are as follows:

1. Insert the IBM DB2 UDB V7.2, Enterprise Edition CD and run Setup.
2. We accepted the default options unless noted as follows:
 - Select **DB2 Enterprise Edition** and **DB2 Application Development Client**.
 - Select **Custom** for the installation type. Refer to the noted documentation for details.
 - We installed DB2 to the c:\ibm\sqlib directory.
 - Create an instance.

Install DB2 V7 FixPak 7 (7.1.0.68)

We installed DB2 V7 FixPak 7 (7.1.0.68), which can be downloaded at:

```
ftp://ftp.software.ibm.com/ps/products/db2/fixes/english-us/db2ntv7/FP7_WR2  
1311/
```

Update JDBC level to JDBC2

Update the JDBC level to JDBC2 as follows:

1. Stop all DB2 Windows services.
2. Run usejdbc2.bat found in the <DB2_HOME>\java12 directory.
3. The inuse file found in the <DB2_HOME>\java12 directory should state the following:

```
JDBC 2.0
```

4. Verify the JDBC functionality (optional).

We have included an IBM site where a JDBC test tool and instructions (jdbcTest.java, jsread2.html) for verification can be downloaded and run on your system. Enter the following FTP link in a Web browser:

<ftp://ftp.software.ibm.com/software/websphere/info/tools/jdbcTest>

5.2.3 IBM HTTP Server installation

This section highlights the key steps for installing and configuring the IBM HTTP Server for use with WebSphere Application Server and WebSphere Commerce.

The IBM HTTP Server installation is organized as follows:

- ▶ Install the IBM HTTP Server
- ▶ Configure the IBM HTTP Server
- ▶ Verify the IBM HTTP Server

Note: For detailed installation instructions, refer to the following:

- ▶ *WebSphere Commerce V5.4 Handbook*, SG24-6567 redbook
- ▶ *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Windows*

Install the IBM HTTP Server

To install the IBM HTTP Server, do the following:

1. Insert the WebSphere Application Server CD, change to the httpd directory and run Setup.
2. We accepted the default options unless noted:
 - We installed the IBM HTTP Server to the c:\ibm\http directory.

Configure the IBM HTTP Server

After the IBM HTTP Server installation is complete, we need to configure the IBM HTTP Server by completing the following steps:

1. Create admin user.
`<HTTP_HOME>\httpasswd -c conf/admin.passwd <http_admin_user>`
2. Enable httpd.conf for SSL.
3. Create IBM HTTP Server key database.
4. Create certificate for the IBM HTTP Server.

Verify the IBM HTTP Server

After the IBM HTTP Server has been configured, verify that the Web server is accessible as follows:

1. Restart the IBM HTTP Server from Windows services.
2. Enter the following URLs in a Web browser:

```
http://<hostname>  
https://<hostname>
```

Note: As part of the procedure for installing the WebSphere Application Server, we will install WebSphere Application Server V4 FixPak 4. In our example, the IBM HTTP Server and WebSphere Application Server are installed on the same node. FixPak 4 includes services fixes for the IBM HTTP Server. After installing FixPak 4, the IBM HTTP Server version is 1.3.19.3.

5.2.4 WebSphere Application Server installation

This section highlights the key steps for installing and configuring the WebSphere Application Server for use with WebSphere Commerce.

The section is organized as follows:

- ▶ Create the WebSphere Application Server repository database
- ▶ Install WebSphere Application Server V4.0.1
- ▶ Install WebSphere Application Server V4 FixPak 5 (V4.0.5)
- ▶ Install WebSphere Application Server V4.0.5 interim fixes
- ▶ Configure the WebSphere Application Server
- ▶ Verify the WebSphere Application Server

Note: For detailed installation instructions, refer to the following:

- ▶ *WebSphere Commerce V5.4 Handbook*, SG24-6567 redbook
- ▶ *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Windows*
- ▶ WebSphere Application Server V4 FixPak 4 (V4.0.4) Readme
- ▶ WebSphere Application Server V4.0.4 interim fix Readme files

Create the WebSphere Application Server repository database

Prior to starting the WebSphere Application Server Administrative server, the WebSphere repository database needs to be created. To create the WebSphere Application Server repository database, complete the following steps:

1. Start a DB2 Command window.

2. Create the WebSphere repository database as follows:

Syntax:

```
db2 create db <db_name>
```

Example:

```
db2 create db was4
```

3. Update the database applheapsz window as follows:

```
db2 update db cfg for was4 using applheapsz 1024
```

Note: The value suggested for the applheapsz in our example is 1024. This is needed by the WebSphere Portal and WebSphere Personalization in the event that you add the WebSphere Application Server Administrative Server node to the same WebSphere Domain as this node.

4. Catalog the database:

```
db2 catalog db was4 as was40
```

5. Connect to the WebSphere database as follows:

Syntax:

```
db2 connect to was40
```

6. List all databases as follows:

```
db2 list db directory
```

Install WebSphere Application Server V4.0.1

To install the WebSphere Application Server V4.0.1, Advanced Edition, do the following:

1. Insert the WebSphere Application Server CD, and run Setup.

2. We accepted the default options unless noted:

- Select **Custom Installation**.

Note: In our example, we manually installed and configured the IBM HTTP Server prior to installing the WebSphere Application Server.

- We installed the WebSphere Application Server to the c:\ibm\was directory.

3. After the installation, rename the <WAS_HOME>\bin\createwasdb.scr file to createwasdb.scr.bak. In our example, we have already manually created the WebSphere Application Server repository database, which is the function of this script.

Install WebSphere Application Server V4 FixPak 5 (V4.0.5)

To install WebSphere Application Server V4 FixPak 5 (V4.0.5), do the following:

1. Ensure you have uninstalled interim fixes before installing FixPak 5.

Note: Refer to the WebSphere Commerce FixPak V5.4.0.5 readme (WC5405_Readme.pdf) for details on how to determine if you have interim fixes installed, and how to remove them.

2. Ensure the following Windows services have been stopped:

- IBM HTTP Server
- IBM WS AdminServer 4.0

3. Download the WebSphere Application Server V4 FixPak 5 (V4.0.5) at:

<http://www.ibm.com/software/webservers/appserv/support.html>

4. Install the WebSphere Application Server V4 FixPak 5.

```
<temp_dir>\install
```

5. Review the FixPak log file to check for errors.

```
<WAS_HOME>\logs\was40_AE_ptf_5.log
```

Install WebSphere Application Server V4.0.5 interim fixes

After the installation of the WebSphere Application Server V4 FixPak 5, install the required interim fixes for WebSphere Commerce as follows:

1. Enter the following URL to access the WebSphere Application Server support page for interim fixes:

<http://www-3.ibm.com/software/webservers/appserv/was/support/>

2. Click **All Fixes, FixPaks and Tools**.

3. Select 4.0.5 from the search criteria list and then click **Submit**.

4. From the interim fixes listing, download the following:

- Cumulative security interim fix
- Cumulative security JSSE interim fix (addresses WebSphere/LDAP SSL problem)
- Cumulative WebSphere plug-in interim fix
- PQ73233, 4.0.5: XML imports and exports fail

5. Set the Java PATH by entering the following in a command window:

```
SET PATH=%PATH%;c:\ibm\was\java\bin
```

Where c:\ibm\was is the directory where the WebSphere Application Server has been installed.

6. Install the interim fix as follows for each interim fix listed above:

```
java -jar <e-fix> -target <WAS_HOME>
```

7. Verify the interim fix has been installed:

- Verify a backup of the interim fix has been created:

```
<WAS_HOME>\e-fix\<efix#>
```

- Verify the product.xml file has been updated:

```
<WAS_HOME>\properties\com\ibm\websphere\product.xml
```

8. Ensure that you have restarted your system after installation.

Configure the WebSphere Application Server

To configure the WebSphere Application Server, do the following:

1. Start the WebSphere Administrative Server (IBM WS AdminServer 4.0 Windows services).
2. Verify the WebSphere Administrative Server has started by reviewing the <WAS_HOME>\logs\tracefile.
3. Add host aliases to the default virtual host from the WebSphere Administrative Console:
 - <hostname>:80
 - <hostname>:443
 - <fully_qualified_hostname>:80
 - <fully_qualified_hostname>:443
4. Regenerate the Web server plug-in.
5. To make the plugin-cfg.xml file updates take effect immediately, restart the IBM HTTP Server and restart the Default Application server.

Verify the WebSphere Application Server

After the WebSphere Application Server, verify the configuration as follows:

1. Ensure the following Windows services have been started:
 - DB2
 - IBM HTTP Server
 - IBM WS Admin Server 4.0
2. From the WebSphere Administrative Console, start the Default Application Server.
3. From a Web browser, enter the following URLs:

```
http://<hostname>/servlet/snoop
```

```
https://<hostname>/servlet/snoop
```

```
http://<hostname>/webapp/examples/showCfg
```

5.2.5 WebSphere Commerce installation

This section highlights the key steps for installing and configuring WebSphere Commerce V5.4, Business Edition and WebSphere Commerce FixPak V5.4.0.5.

The section is organized as follows:

- ▶ Install WebSphere Commerce V5.4
- ▶ Install WebSphere Commerce FixPak V5.4.0.5
- ▶ Post-install WebSphere Commerce FixPak V5.4.0.5

Note: For detailed installation instructions, refer to the following:

- ▶ *WebSphere Commerce V5.4 Handbook*, SG24-6567 redbook
- ▶ *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Windows*
- ▶ *Installation Guide, IBM WebSphere Commerce FixPak V5.4.0.5* (WC5405_Readme.pdf)

Install WebSphere Commerce V5.4

To install WebSphere Commerce V5.4, Business Edition, do the following:

1. Insert the WebSphere Commerce V5.4 CD1 and run **Setup**.
2. We accepted the default options unless noted as follows:
 - When selecting components that will be running on this node (WebSphere Commerce node), in a single-tier configuration, select **The commerce server, a Web server, and a database**.

Note: This is not asking what components to install, but what components will be running on the server. In our example, we have installed the prerequisite components with a higher FixPak level prior to installing WebSphere Commerce.

- We installed WebSphere Commerce in the c:\ibm\wc directory.

Note: Shortly after inserting WebSphere Commerce CD2, the installer will attempt to install WebSphere Application Server V4 interim fixes. These interim fixes are out of date and will not be installed since the level of WebSphere Application Server is V4.0.5.

Install WebSphere Commerce FixPak V5.4.0.5

To install the WebSphere Commerce FixPak V5.4.0.5, do the following:

1. Stop the following Windows services:
 - IBM HTTP Server
 - IBM WS AdminServer 4.0
 - IBM WC Configuration Manager Server
2. Ensure that you have installed the prerequisite WebSphere Application Server V4 FixPak 5 and interim fixes (refer to 5.2.4, “WebSphere Application Server installation” on page 177).
3. Download the FixPak from the following URL:
4. Unzip the WebSphere Commerce FixPak V5.4.0.5 zip file and enter the following from a command prompt to start the FixPak installer:
5. Follow the FixPak installation prompts.
6. Review the log file for errors.

```
<WC_HOME>\service\fixpak\5403\MCFixpak5405_wc_WIN.log
```

Post-install WebSphere Commerce FixPak V5.4.0.5

After the WebSphere Commerce FixPak V5.4.0.5 has been installed, there are several configurations steps that need to be completed. In our example, we are using WebSphere Application Server V4.0.5 and interim fixes. Refer to the following technote for running WebSphere Commerce FixPak V5.4.0.5 on WebSphere Application Server V4.0.3 (applies to V4.0.5).

```
http://www.ibm.com/support/docview.wss?rs=494&uid=swg21054765
```

This section describes how to address the issues related to the MassLoad utility documented in the technote.

1. Add the JITC_COMPILEOPT environment variable as follows to either the Windows system environment or adminserver.bat:
 - Windows system environment
 - Variable: JITC_COMPILEOPT
 - Value:
JITC_COMPILEOPT=SKIP{com/ibm/wca/MassLoader/Formatter/JDBCFormatter\$FormatWorker}{run}

or

```
- adminserver.bat (all on one line)
SET
JITC_COMPILEOPT=SKIP{com/ibm/wca/MassLoader/Formatter/JDBCFormatter$Form
atWorker}{run}
```

Note: In our case, we updated both the Windows environment variable and the adminserver.bat. We started the WebSphere Application Server Administrative Server using the adminserver.bat from this point forward. We updated the Windows environment as well to avoid problems running the MassLoader from the command line (Java utility to load XML data).

Note: Stopping the WebSphere Administrative Server

To stop the WebSphere Administrative Server in a controlled manner, especially when starting using adminserver.bat, we recommend using the WebSphere Control Program (wscp). For example:

1. Change to the <WAS_HOME>\bin directory
2. Enter wscp at the command line to start the WebSphere Control Program.
3. Enter the following command to stop the WebSphere Administrative Server from the wscp command line.

```
Node stop /Node:wcserv1/
```

Where wcserv1 is the node name.

4. Horizontal cloning.

Note: Do not use this workaround if using the IBM Commerce Enhancement Pack - April 2003 Edition (see note box below).

When configuring horizontal cloning with WebSphere Application Server V4, the enterprise application name and enterprise application directory in <WAS_HOME>\InstalledApps need to be the same. To work around a problem with the default <WC_HOME>\xml\config\was.deployed.EJB.xml, it has been recommended that the was.deployed.EJB.xml file be modified as follows:

```
<enterprise-application action="create" name="WC Enterprise App
${DRIVER_INSTANCE_NAME}$"> )
```

Note: At the time of writing this redbook, performing the above stated workaround for horizontal cloning causes the ImportEJB.bat (CMRedeploy) for the IBM Commerce Enhancement Pack - April 2003 Edition to fail in subsequent steps. It appears that the enterprise application name "WebSphere Commerce Enterprise Application" is hardcoded in CMRedeploy. Performing the horizontal cloning workaround renames the enterprise application name to "WC Enterprise App" to match the directory. At the time of writing this redbook, a workaround to this problem was not known.

Check for updates on the IBM support site for using IBM Commerce Enhancement Pack - April 2003 Edition and horizontal cloning. We advise the reader to check the WebSphere Commerce support page for the most current information on workarounds or interim fixes:

http://www-3.ibm.com/software/webservers/commerce/wc_be/support.html

5. Update the <WC_HOME>\bin\setenv.bat (workaround for loader property file not being found) to include the following:

```
set JAVA_EXE=%JAVA_HOME%\bin\java -Xbootclasspath/p:c:\ibm\wc\lib\jlog.jar
```

Note: In order for the WebSphere Commerce <instance> application server to recognize this change, the application server will need to be restarted. In our example, we have not yet created the WebSphere Commerce <instance> application server.

6. Update WebSphere Commerce Staging Server.

If you are using the WebSphere Commerce Staging Server feature, and have an existing DB2 WebSphere Commerce instance database, you will need to manually update the database by running the FixPak version of the wcs.stage.trigger.sql script found in the <WC_HOME>\schema\db2 directory.

For example, enter the following from a DB2 command window:

```
db2 connect to <wc_db> user db2admin using <password>
db2 -tvf wcs.stage.trigger.sql
```

Note: If you have not created a WebSphere Commerce instance or database yet, it is not necessary to update the database. The FixPak installer has updated the wcs.stage.trigger.sql file and it will be used when the WebSphere Commerce instance and database are created.

7. Copy search.xml to WebSphere Commerce instance XML directory.

If you have created a WebSphere Commerce instance before the WebSphere Commerce FixPak V5.4.0.5 installation, you will need to manually copy the search.xml file now found in the <WC_HOME>\instances\default\xml directory to the <WC_HOME>\instances\<instance_name>\xml directory for each instance you have created.

Note: Once the WebSphere Commerce FixPak V5.4.0.5 is installed, the default instance directory will contain the search.xml file and all newly created WebSphere Commerce instances will contain this file.

8. Enable double-click (optional).

Refer to the *Installation Guide, IBM WebSphere Commerce FixPak V5.4.0.5*.

5.2.6 WebSphere Commerce Payments installation

This section highlights the key steps for installing WebSphere Commerce Payments. After the release of WebSphere Commerce V5.4, WebSphere Commerce Payments FixPak 3.1.3 was made available for download. The FixPak contains fixes and changes the name of the product to WebSphere Commerce Payments.

The section is organized as follows:

- ▶ Pre-requisites for WebSphere Payment Manager V3.1.2
- ▶ Install WebSphere Payment Manager V3.1.2
- ▶ Install WebSphere Commerce Payments FixPak V3.1.3.1

Note: For detailed installation instructions, refer to the following:

- ▶ *WebSphere Commerce V5.4 Handbook*, SG24-6567
- ▶ *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Windows*
- ▶ *Installation Guide, IBM WebSphere Payment Manager for Multiplatforms*

Pre-requisites for WebSphere Payment Manager V3.1.2

Prior to installing WebSphere Payment Manager V3.1.2, do the following:

1. Start a DB2 command window.
2. Create WebSphere Commerce Payments database.

```
db2 create db pay1
```

3. Update applheapsz.

```
db2 update db cfg for pay1 using applheapsz 256
```

4. Catalog the WebSphere Commerce Payments database.

```
db2 catalog db pay1 as payman
```

Note: When using the catalog database DB2 command, the <node_name> parameter is optional if the database is local.

If you use the <node_name> parameter when performing the catalog database operation, ensure you use the proper case of the <node_name> as known within DB2. If you are not sure, view the node name from the DB2 Control Center. If the database is local, using the <node_name> parameter is optional.

Where <node_name> is the TCP/IP node cataloged in the configuration of the WebSphere repository database discussed in "Create the WebSphere Application Server repository database" on page 177.

5. Verify the connection to the database:

```
db2 connect to payman user <db2admin_user> using <db2admin_password>
```

6. Ensure the following are started:

- DB2
- WebSphere Administrative Server

WebSphere Administrative Server is started depending on how you implemented the environment variable for JIIC_COMPILEOPT:

- adminserver.bat
- or
- IBM WS AdminServer 4.0 (Windows services)

Install WebSphere Payment Manager V3.1.2

To install WebSphere Payment Manager V3.1.2, do the following:

1. Insert the WebSphere Payment Manager V3.1.2 CD and run Install.
2. We accepted the default options, except the following:
 - We installed to the c:\ibm\wcpay directory.
 - We named the database payman.

Install WebSphere Commerce Payments FixPak V3.1.3

To install the WebSphere Commerce Payments FixPak V3.1.3, do the following:

1. Download the FixPak from the following URL:
`https://www6.software.ibm.com/dl/paymgr/srvupdts-p`
2. Ensure the WebSphere Application Server Administrative Server is started (adminserver.bat or IBM WS AdminServer 4.0 Windows services).
3. Ensure the WebSphere Payment Manager application server is stopped (use the WebSphere Administrative Console or the command line to stop the application server).
4. Make sure the directory containing the ptf.class is in the CLASSPATH environment variable. For example:

```
set CLASSPATH=%CLASSPATH%;.
```

5. Set the PATH environment variable for Java. For example:

```
set PATH=%PATH%;c:\ibm\was\java\bin
```

6. To start the PTF installer, enter the following command with no extension:

```
java <ptf_filename>
```

Where <ptf_filename> is the PTF file name without the .class extension.

Note: The PTF class file must be in a directory that is *not* where WebSphere Commerce Payments is installed and that has write access.

Install WebSphere Commerce Payments FixPak V3.1.3.1

Refer to the following URL for a technote about WebSphere Commerce Payments V3.1.3.1. In our example, we need FixPak V3.1.3.1 for the WebSphere Commerce V5.4.0.5 and WebSphere Application Server V4.0.5.

```
http://www-1.ibm.com/support/docview.wss?rs=494&context=SSZLC2&q=&uid=swg21106207&loc=en\_US&cs=utf-8&lang=en
```

Note: Refer to the WebSphere Commerce Payments FixPak V3.1.3.1 Readme for detailed instructions and important notes before installation.

There is mention of additional instructions (this appears to be for AIX/DB2).

To install WebSphere Commerce Payments FixPak V3.1.3.1, do the following:

1. Download the FixPak from the following URL (requires that you are registered):

```
https://www6.software.ibm.com/dl/paymgr/priv/srvupdts-h?S\_PKG=ctr31ww
```

2. Ensure the WebSphere Application Server Administrative Server is started (adminserver.bat or IBM WS AdminServer 4.0 Windows services).
3. Ensure the WebSphere Commerce Payments application server is stopped (use the WebSphere Administrative Console or the command line to stop the application server).
4. Make sure the directory containing the ptf.class is in the CLASSPATH environment variable. For example:

```
set CLASSPATH=%CLASSPATH%;.
```
5. Set the PATH environment variable for Java. For example:

```
set PATH=%PATH%;c:\ibm\was\java\bin
```
6. To start the PTF installer, enter the following command with no extension:

```
java <ptf_filename>
```

Where <ptf_filename> is the PTF filename without the .class extension.

Note: The PTF class file must be in a directory that is *not* where WebSphere Commerce Payments is installed and that has write access.

5.2.7 Commerce Enhancement Pack installation

This section highlights the key steps for installing IBM Commerce Enhancement Pack - April 2003 Edition. The install is comprised of two parts:

- CEP October (WIN_APRI2003_CommerceEnhancementPack.zip)
- CEP April cumulative interim fix
(WIN_02-10-2003_cumulative_OCT2002_CommerceEnhancementPack.zip)

Note: For detailed installation instructions, refer to the *Getting Started, IBM Commerce Enhancement Pack* guide included with the IBM Commerce Enhancement Pack - April 2003 Edition.

Commerce Enhancement Pack - October installation

To install the IBM Commerce Enhancement Pack - April 2003 Edition, base components from the CommerceEnhancementPack_WIN_OCTOBER2002.zip that was included in the WIN_APRI2003_CommerceEnhancementPack.zip.

Note: The packaging of the IBM Commerce Enhancement Pack - April 2003 Edition can be very confusing. We have attempted to explain what we did.

1. Back up the WebSphere Commerce instance database if one exists at this stage from a previous instance being created.
2. Download the Windows version of the IBM Commerce Enhancement Pack (WIN_APRI2003_CommerceEnhancementPack.zip) to a temporary folder from the following URL:

<http://www.ibm.com/software/webservers/commerce/epacks/v54/>
3. Navigate to the temporary folder and unzip WIN_APRI2003_CommerceEnhancement.zip.

You should now see:
 - CommerceEnhancementPack_WIN_OCTOBER2002.ZIP
 - WC_3D_Secure.zip
4. Unzip the CommerceEnhancementPack_WIN_OCTOBER2002.zip.

For example, we unpacked the zip file to c:\temp\cep. We will refer to the unzip directory as <CEP_HOME>.
5. Start the installation by executing Setup.
6. Follow the prompts to install and configure the IBM Commerce Enhancement Pack.

Note: When running the WebSphere Commerce IC Checker to ensure that your WebSphere Commerce machine is configured properly, all error messages can be safely ignored.

7. Check the <WC_HOME>\logs\EPInstall.log file for errors to ensure that the IBM Commerce Enhancement Pack has installed properly:

Commerce Enhancement Pack - cumulative interim fix installation

To install the Commerce Enhancement Pack cumulative fix, do the following:

1. Download the Windows version of the IBM Commerce Enhancement Pack cumulative fix:

WIN_02-10-2003_cumulative_OCT2002_CommerceEnhancementPack.zip

to a temporary folder from the following URL:

<http://www.ibm.com/software/webservers/commerce/epacks/v54/>
2. Unzip
WIN_02-10-2003_cumulative_OCT2002_CommerceEnhancementPack.zip
(IBM Commerce Enhancement Pack - April 2003 Edition). For example, we

unpacked the zip file to c:\temp\cep\fix. We will refer to this unzip directory as <CEP_HOME>\cepfix.

3. Start the installation by running `install_wc.bat`.

Note: Refer to the readme.htm included with the cumulative fix. If you have previously created a WebSphere Commerce instance, you will need to run `install_was.bat`.

4. Follow the prompts to install and configure the IBM Commerce Enhancement Pack.

Note: When running the WebSphere Commerce IC Checker to ensure that your WebSphere Commerce machine is configured properly, all error messages can be safely ignored.

5. Check the <WC_HOME>\logs\EPinstall.log file for errors to ensure that the IBM Commerce Enhancement Pack has installed properly:

5.2.8 WebSphere Commerce instance creation

This section highlights the key steps for creating a WebSphere Commerce instance using the Configuration Manager.

Note: For detailed installation instructions, refer to the following:

- ▶ *WebSphere Commerce V5.4 Handbook*, SG24-6567 redbook
- ▶ *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Windows*

To create a WebSphere Commerce instance, do the following:

1. Ensure the following Windows services are started:
 - DB2
 - IBM WC Configuration Manager
 - IBM WS AdminServer 4.0
2. Ensure the IBM HTTP Server Windows services is stopped.

This is done as a workaround to avoid an instance creation failure when the IBM HTTP Server is configured with SSL support prior to creating an instance.

3. Create a WebSphere Commerce instance using the Configuration Manager, which can be launched by clicking **Start -> Programs -> WebSphere Commerce -> Configuration**.
4. The default logon ID and password are webadmin/webibm. You will be prompted to change the password.
5. We accepted the default options unless otherwise noted.
 - Instance name: wc1
 - Database name: wc1db

Note: Staging Server

If you are planning on using this node as a staging server and using the WebSphere Commerce staging utilities, ensure the **Use Staging Server** check box is checked within the Configuration Manager Instance Creation wizard on the Database tab.

If this is not a staging server or you are not planning on using the WebSphere Commerce staging utilities, do not check Use Staging Server.

6. Verify the instance creation.

Review the log files found in the <WC_HOME>\instances\<instance>\logs directory.

5.2.9 Commerce Enhancement Pack post-install configuration

This section highlights the key steps for post-install configuration of the IBM Commerce Enhancement Pack - April 2003 Edition

Note: For details, refer to the *Getting Started, IBM Commerce Enhancement Pack* guide included with the IBM Commerce Enhancement Pack - April 2003 Edition.

In our example, we created an instance after the Enhancement Pack was installed. For this reason, we need to complete some steps in this section.

1. Rebind the data sources after the instance creation.

After the instance has been created, it may be necessary to run the following command to rebind the data sources (workaround for blank page for tools):

```
<WAS_HOME>\bin\wscp -f rebindDataSources.tcl
```

Note: WebSphere Application Server Administrative Server must be running prior to running the **wscp** command.

2. Update the WebSphere Commerce instance database.
 - a. Execute the EP1_DBUpdate.db2.bat script found in the <WC_HOME>\EnhancementPack\bin directory as follows from a DB command window:

```
EP1_DBUpdate.db2 <wc_dbname> <dbuser> <dbpassword>
```

- b. To verify the DB2 script worked properly, review the <WC_HOME>\logs\EPDBUpdate.log file.

Note: As documented in the *Getting Started, IBM Commerce Enhancement Pack* guide, several errors will be listed in the EPDBUpdate.log for cases where the EP1_DBUpdate.db2.bat is run on an instance created after the IBM Commerce Enhancement Pack - April 2003 Edition is installed.

3. Ensure that the WebSphere Administrative Server is started (adminserver.bat), and that the WebSphere Commerce - <instance> application server is stopped.
4. We recommend that you create a backup of the WebSphere Commerce Enterprise Application prior to importing the new EJBs in the next step.
 - a. Start the WebSphere Application Server Administration Console.
 - b. Select and expand the **WebSphere Application Domain -> Enterprise Applications**.
 - c. Select **WebSphere Commerce Enterprise Application - <instance_name>**, right-click and select **Export Application**.
 - d. Follow the on-screen instructions, and export the application to another location for backup. We created a directory c:\ibm\wc\wcentapp.bak and exported the enterprise application to this directory.
5. Import new EJBs.
 - a. If you are using adminserver.bat to start the WebSphere Administrative Server, you will temporarily need to stop the server. The ImportEJB.bat does not recognize that the WebSphere Administrative Server is running when started using adminserver.bat.

```
<WAS_HOME>\bin\wscp  
Node stop /Node:wcserv1/
```

- b. Start the IBM WS AdminServer 4.0 Windows services.

- c. Run the following command from a Windows command prompt:

```
<WC_HOME>\EnhancementPack\bin\ImportEJB -instanceName <instance_name>  
<database_type>
```

Where *<instance_name>* is the name of your WebSphere Commerce instance, and *<database_type>* is either DB2 or Oracle.

For example:

```
<WC_HOME>\EnhancementPack\bin\ImportEJB -instanceName wc1 DB2
```

- d. Verify the ImportEJB.bat worked properly by reviewing the
<WC_HOME>\logs\CMRRedeploy_EJB.log for errors.
6. If you redeploy EJBs into the WebSphere Commerce Enterprise Application, you may have to follow the instructions in technote 1066544 found at:
http://www.ibm.com/software/webservers/commerce/wc_be/support.html

Note: We did not need to perform this step for the configuration and procedure we have documented.

7. Other installable options

There are many other options that can be configured such as Rules-based Discounts, Web services and Guided Sell. These options are not needed for the ITSO B2B CEP store, so we did not configure them.

Refer to the *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition* for detailed instructions.

8. Stop the IBM WS AdminServer 4.0 Windows services, and start the WebSphere Administrative Server using the updated `adminserver.bat`.

5.2.10 WebSphere Commerce administration tools verification

This section describes how to verify that the WebSphere Commerce administration tools are working properly.

To verify the WebSphere Commerce administration tools, do the following:

1. Ensure the following are started:
 - DB2 Server Windows services
 - IBM HTTP Server Windows services
 - WebSphere Administrative Server (`adminserver.bat`)
2. From the WebSphere Administration Console, start the WebSphere Commerce *<instance>* application server.
3. Verify WebSphere Commerce Store Services:

- a. Start Store Services as follows:
`https://<hostname>:8000/storeservices`
or
From Windows, select **Start -> Programs -> IBM WebSphere Commerce -> Store Services**.
- b. Verify that you can log on.
 - User ID: wcsadmin
 - Password: wcsadmin (the default)

At this stage, we want to make sure that the tool is able to start without errors.
4. Verify the WebSphere Commerce Administration Console:
 - a. Start the WebSphere Commerce Administration Console as follows:
`https://<hostname>:8000/adminconsole`
or
From Windows, select **Start -> Programs -> IBM WebSphere Commerce -> Administration Console**.
 - b. Verify that you can log on.
 - User ID: wcsadmin
 - Password: <password>
5. Verify WebSphere Commerce Accelerator:
 - a. Start the WebSphere Commerce Accelerator as follows:
`https://<hostname>:8000/accelerator`
or
From Windows, select **Start -> Programs -> IBM WebSphere Commerce -> WebSphere Commerce Accelerator**.
 - b. Verify that you can log on.
 - User ID: wcsadmin
 - Password: <password>

5.2.11 WebSphere Commerce Payments configuration

This section describes the necessary steps to configure the WebSphere Commerce Payments server.

If WebSphere Commerce Payments is started when the store is published, the payment server is configured during the store publishing process. If not, start

WebSphere Commerce Payments and republish the store or manually configure WebSphere Commerce Payments.

Configure WebSphere Commerce Payments

After installing WebSphere Commerce FixPak V5.4.0.5 and if WebSphere Commerce Payments is installed on your local WebSphere Commerce machine, you must follow the steps below to configure a WebSphere Commerce administrator ID to use with WebSphere Commerce Payments.

Note: Refer to the *Installation Guide, IBM WebSphere Commerce FixPak V5.4.0.5* for more information.

Start WebSphere Commerce Payments

To start WebSphere Commerce Payments, do the following:

1. Start the following Windows services:
 - DB2
 - IBM HTTP Server
 - WebSphere Administrative Server (adminserver.bat)
2. Start the WebSphere Commerce Payments application server from the WebSphere Administrative Console.
3. Start the WebSphere Commerce Payments Server for the <WCPAY_HOME> directory from the command line as follows:

IBMPayServer
4. When prompted, enter the database password for the payman database.

Verify the WebSphere Commerce Payments Admin Console

To verify the WebSphere Commerce Payments Administration Console, do the following:

1. Start the WebSphere Commerce Payments Administration Console as follows:
 - `http://<hostname>/webapp/PaymentManager`or
 - From Windows, select **Start -> Programs -> WebSphere Commerce Payments -> Commerce Payments Logon**.
- c. Verify that you can log on.
 - User ID: wcsadmin
 - Password: <password>

2. The WebSphere Commerce Payments Administration Console can be used to set up additional methods of credit card payment (Visa, American Express), add payment cassettes (SET, Cyber Cash, etc.), create accounts, approve orders, etc.

5.2.12 Database backup

Before publishing a store, we recommend that you back up the WebSphere Commerce instance database. For details, refer to "Back up a DB2 database" on page 375.

5.2.13 Enable WebSphere Commerce portal adapter

To enable the WebSphere Commerce portal adapter, do the following:

Note: For more information, refer to the *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*.

1. Stop the WebSphere Commerce <instance> application server from the WebSphere Application Server Administration Console.
2. During the IBM Commerce Enhancement Pack - April 2003 Edition installation, the <WC_HOME>\schema\db2\wcs.portal.sql script is executed by the EP1_DBUpdate.db2.script. This .sql script updates the WebSphere Commerce instance database tables PVCDEVMDL, PVCDEVSPEC, and PVCMDLSPEC for the Portal and WAP Portal adapter.

By default, wcs.portal.sql script sets the deviceFormatId to the next available entry number. You must ensure the MODEL_ID in the PVCDEVMDL table updated by the wcs.portal.sql script matches the deviceFormatId in the instance XML file (see Example 5-1 on page 197).

Ensure the wcs.portal.sql file has been executed. When installing the IBM Commerce Enhancement Pack - April 2003 Edition, this .sql file is executed by the EP1_DBUpdate.db2.script. If you have executed the EP1_DBUpdate.db2.script as documented in the installation procedure, this step has already been completed.

3. Determine the MODEL_ID in the PVCDEVMDL table for the adapter (PORTAL and WAP PORTAL). For example, to query the database for the PORTAL adapter, enter the following from a DB2 command window:

```
db2 connect to <wc_dbname> user <db2_admin> using <password>
db2 select * from pvcdevmdl
```

Record the value of the MODEL_ID for the desired adapter.

Note: You may consider piping the output to a text file for recording purposes. For example:

```
db2 select * from pvcdevmd1>c:\temp\model_id.out
```

4. Change to the directory of the WebSphere Commerce <instance>.xml file:

```
<WC_HOME>\instances\<instance_name>\xml
```

5. Back up the original <instance>.xml file to <instance>_org.xml.
6. Update the WebSphere Commerce <instance>.xml file found in the <WC_HOME>\instances\<instance>\xml directory with a text editor to enable the WebSphere Commerce Portal adapter.
 - a. Update deviceFormatId.
 - i. Search deviceFormatId within the HttpAdapter <tag> for PORTAL in the <instance>.xml file.
 - ii. Update the value of the deviceFormatId to match the MODEL_ID recorded for the adapter.
 - b. Enable the adapter.

Enable the deployment descriptor of the WebSphere Commerce portal adapter by changing the enabled="false" value to enabled="true" in the <instance>.xml file as seen in Example 5-1.

- c. Save the <instance>.xml file.

Example 5-1 Enable the WebSphere Commerce Portal adapter

```
<HttpAdapter deviceFormatId="-3"
  deviceFormatType="PVCAdapter"
  deviceFormatId="-1"
  enable="true"
  factoryClassname="com.ibm.commerce.portaladapters.HttpPortalBrowserAdapter"
  name="PORTAL">
<PVCAdapter bufferTimeout="5"preferredLogonTimeout="20"registrationMode="0"/>
</HttpAdapter>
```

When complete with the changes to the <instance>.xml file, save the file and restart the WebSphere Commerce <instance> application server.

5.2.14 Disable the WebSphere Commerce cache

When using the Portal adapter and the WAP Portal adapter, the JSP directory includes JSPs with the same name as the standard HTTP adapter clients. The WebSphere Commerce cache cannot differentiate the JSPs that have been

cached for the different devices (adapters). This may cause a problem when rendering the JSPs for the requesting device type. For this reason, the WebSphere Commerce cache must be disabled when more than one of the following adapters is used:

- HTTP adapter
- Portal adapter
- WAP Portal adapter

To disable the WebSphere Commerce cache, do the following:

1. Start the WC Configuration Manager Server Windows services.
2. Start the WebSphere Commerce Configuration Manager.
3. Log on to the Configuration Manager.
4. From the left pane, select and expand **WebSphere Commerce** -> **<node>** -> **Caching Subsystem**.
5. Uncheck the **Enable Cache** check box and click **Apply**.
6. Select **Components** -> **CacheDaemon**.
7. Uncheck the **Enable Component** check box and click **Apply**.
8. Exit Configuration Manager.
9. Restart the WebSphere Commerce <instance> application server.

You have now completed the base configuration of the WebSphere Commerce node. If you are using VMWare or Ghost (test purposes), this is an appropriate time to create an image.

5.2.15 Commerce Enabled Portal - APAR JR18068

The following URL includes a link to a commerce enabled portal APAR JR18068. This needs to be installed on top of the IBM Commerce Enhancement Pack - April 2003 Edition cumulative fix.

1. Enter the following URL:
http://www-1.ibm.com/support/docview.wss?rs=494&context=SSZLC2&q=4.2&uid=swg21107559&loc=en_US&cs=utf-8&lang=en
2. Download WIN_JR18068.zip and *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition* PDF from the bottom of the page.
3. Unzip the WIN_JR18068.zip to a temp directory (for example, c:\temp\cepf\cepfix).
4. Refer to the Readme included with the zip for installation instructions.

5.3 Directory Server (LDAP) node implementation

The IBM Directory Server V4.1 server is used by the WebSphere Portal Server for authentication and user lookups and is used by the WebSphere Commerce node as a user registry and for authentication. When configuring both nodes for single sign-on, the directory is shared among the participating applications, and users only have to log on one time. In our scenario, the users will authenticate from the WebSphere Portal, which is configured to use IBM SecureWay Directory, and a LtpaToken will be passed to the single sign-on participating application, WebSphere Commerce.

This section describes the high-level steps to install the IBM Directory Server node within the ITSO test environment. In this example, the IBM Directory Server is not dependent on other nodes. The DB2 Server is installed on the IBM Directory Server node. Since authentication will be routed through the IBM Directory Server, we did not want to have additional network traffic to retrieve each user authentication from a remote database server.

Note: More detailed information can be found in the following:

- ▶ *WebSphere Commerce V5.4 Handbook*, SG24-6567 redbook
- ▶ *Commerce Enabled Portal Integration Guide*, IBM Commerce Enhancement Pack April 2003 Edition
- ▶ *IBM WebSphere Portal V4.1 Handbook*, SG24-6883 redbook

This high-level implementation steps for the IBM Directory Server are as follows:

1. Windows 2000 Server installation
2. DB2 Server installation
3. IBM HTTP Server installation
4. IBM Directory Server installation
5. IBM Directory Server configuration
6. IBM Directory Server verification
7. Import data for WebSphere Portal and WebSphere Commerce
8. Configure WebSphere Commerce for LDAP
9. Enable SSL between WebSphere Commerce and LDAP

5.3.1 Windows 2000 Server installation

In preparation for the installation of DB2, ensure the following tasks have been completed:

1. Install Windows 2000 Server and Windows 2000 Service Pack 3.

2. Ensure an administrator user is logged in for installation of the WebSphere Commerce and supporting software with the following user rights:
 - Act as part of the operating
 - Create a token object
 - Increase quotas
 - Log on as a service
 - Replace a process level token
3. Install Internet Explorer 5.5 and service pack or higher.
4. Verify the configuration of the TCP/IP network (host name, IP address).

5.3.2 DB2 Server installation

For details on installing and configuring the DB2 Server, refer to 5.7, "DB2 Server node implementation" on page 245. The following tasks need to be completed:

- ▶ Install DB2 UDB V7.2 Enterprise Edition
- ▶ Install DB2 V7 FixPak 7 (7.1.0.68)
- ▶ Update JDBC level to JDBC2

5.3.3 IBM HTTP Server installation

For details on installing and configuring the IBM HTTP Server, refer to 5.2.3, "IBM HTTP Server installation" on page 176. The following tasks need to be completed:

- ▶ Install the IBM HTTP Server

Note: In order to apply the server level to the IBM HTTP Server included with the WebSphere Application Server V4, FixPak 5, we had to manually copy the <WAS_HOME>\java directory from the WebSphere Portal node to fulfill the FixPak installer requirement (Java program).

- ▶ Configure the IBM HTTP Server
- ▶ Verify the IBM HTTP Server

5.3.4 IBM Directory Server installation

WebSphere Portal V4.2.1 and commerce enabled portals provided in the IBM Commerce Enhancement Pack - April 2003 Edition use IBM Directory Server V4.1 for a single sign-on configuration with WebSphere Commerce.

IBM Directory Server V4.1 is included with the WebSphere Portal V4.2.1 Enable offering.

Note: For detailed information on installing and configuring the IBM Directory Server V4.1, refer to the following:

- ▶ *WebSphere Commerce V5.4 Handbook*, SG24-6567 redbook
- ▶ *WebSphere Portal V4.1, Windows 2000 Installation*, REDP3593 redpaper
- ▶ *IBM SecureWay Directory V3.2.2 Installation Guide* product guide

To install the IBM Directory Server V4.1, do the following:

1. Ensure the IBM HTTP Server is stopped.
2. IBM Directory Server V4.1 is included with the WebSphere Portal V4.2.1 Enable offering. It can also be downloaded from the following URL:
<http://www.ibm.com/software/network/directory/downloads/>
3. To install IBM Directory Server V4.1, insert the CD, run Setup from the <CD_drive>\win15mp directory.

Note: The installation directory may vary depending on the distribution source of IBM Directory Server.

4. We entered c:\ibm\ldap for the installation directory and then clicked **Next**.
5. When the Language window appears, we selected **English** and then clicked **Next**.
6. When prompted for the Setup type, we selected **Typical** and then clicked **Next**.
7. When prompted for the Features to Install, we checked the following and then clicked **Next**:
 - Client SDK 4.1
 - DMT 4.1 & Java
 - Server 4.1
8. When the Select Program Folder window appears, we accepted the default and clicked **Next**.
9. When prompted to enter the Web server configuration file for the IBM HTTP Server, we entered c:\ibm\http\conf\httpd.conf and then clicked **Next**.
10. When the new LDAP administrator Distinguished Name and Password window appears, we entered the following and then clicked **Next**:
 - Administrator Distinguished Name: cn=root
 - Administrator password: <password>
 - Confirm password: <password>

11. Click **Next** to begin copying files.
12. When prompted about beginning GSKit installation in the background, click **OK**.
13. When prompted, review the readme and then click **Next**.
14. After the installation is complete, select **Yes**, restart my system and then click **Next** and then click **Finish**.

5.3.5 IBM Directory Server configuration

After the installation, it is necessary to configure the directory server. The following sections cover the initial setup of the directory server using the IBM Directory Server V4.1 configuration utility:

- ▶ Create the directory administrator user
- ▶ Create the directory database
- ▶ Configure the Web server for directory administration

Note: This configuration can be performed during the installation process or after installation, as is the case in our example. It is assumed that none of the three options has been configured. If any of the three steps has been set up during the installation process, do not check the option when prompted.

1. Ensure the IBM HTTP Server is stopped.
2. Launch IBM Directory Server by clicking **Programs -> IBM Directory Server 4.1 > Directory Configuration**.
3. Select the three available options as seen in Figure 5-2 on page 203 and then click **Next**.

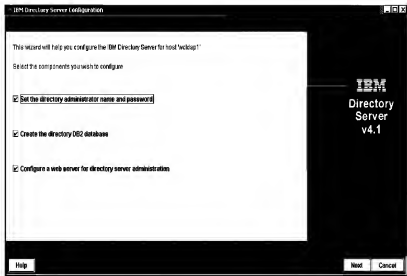


Figure 5-2 IBM Directory Server V4.1 configuration options

4. Enter the Administrator DN and password, and then click **Next**. We recommend that you keep the default Administrator DN, `cn=root`.
5. When the DB2 Database window appears, select **Create a default LDAPDB2 database**, then click **Next**.
6. Select **Create a Universal DB2 database (UTF-8)** when prompted, and then click **Next**.
7. Select the drive where you want to create the database, and click **Next**.
8. A new window to configure the Web server appears. Select **IBM HTTP Server** and then click **Next**.
9. Enter the full path name of the Web server configuration file. By default, the configuration tool suggests a file path. Ensure it is correct and change it if needed. Click **Next**.
10. A Configuration Summary window appears as shown in Figure 5-3 on page 204. Review the values entered and click **Configure** to start the configuration process.

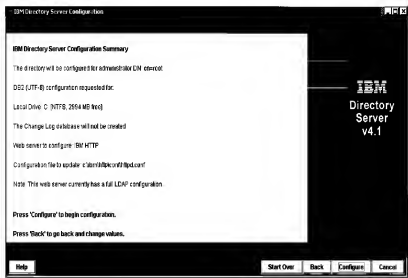


Figure 5-3 IBM Directory Server V4.1 configuration summary window

11. When the Configuration Summary window re-appears, review the results and then click **OK**.

5.3.6 IBM Directory Server verification

After installing and configuring the IBM Directory Server, we recommend that you verify the functionality of the server.

Starting services

The following services must be started for IBM Directory Server to be managed.

1. Start the IBM HTTP Server.

This can be done in Windows services or from the command line as follows:

```
c:\> net stop "IBM HTTP Server"  
c:\> net start "IBM HTTP Server"
```

2. Start the IBM Directory Server V4.1 server.

This can be done in Windows services or from the command line as follows:

```
c:\> net start "IBM Directory Server V4.1"
```

Ensure the service "DB2 - LDAPDB2" is also started. To list the services running, type:

```
c:\net start
```

If service "DB2 - LDAPDB2" is not running, type:

```
c:\net start "DB2 - LDAPDB2"
```

Verify the administration tools

Verify the following IBM Directory Server administration tools:

1. Start the IBM Directory Server Administration Web-based tool.
 - a. Enter the following URL in a Web browser:
`http://<ldap_server_hostname>/ldap`
 - b. Log on as follows:
 - Admin ID: cn=root
 - Password: <password>
2. Start the IBM Directory Server - Directory Management Tool (DMT).
To launch the DMT, select **Start -> Programs -> IBM Directory Server -> Directory Management Tool**.

Note: For more detailed information on verifying the IBM SecureWay Directory, refer to the *WebSphere Commerce V5.4 Handbook*, SG24-6567.

5.3.7 Import data for WebSphere Portal and WebSphere Commerce

Now that IBM Directory Server is installed, configured and the administration tools have been verified, the directory server needs to be configured for WebSphere Portal and WebSphere Commerce.

This section includes the following:

- ▶ Create a new suffix
- ▶ Import WebSphere Portal LDIF
- ▶ Import WebSphere Commerce LDIF
- ▶ DMT configuration for WebSphere Commerce

Create a new suffix

Before data can be imported into the IBM Directory Server database, a suffix must exist.

1. Ensure the IBM Directory Server service is started.

2. Start the IBM Directory Server Administration Web-based tool by entering the following URL in a Web browser:

`http://<hostname>/ldap`

3. When the Administration Logon windows appears, enter the following:
 - Admin ID: `cn=root`
 - Password: `<password>`
4. From the left navigation page, select and expand **Settings -> Suffixes**.
5. Enter the name of the suffix DN.

This is the suffix DN that will be contained in the LDAP Data Interchange Format (LDIF) file used to load data as follows and then click **Update**. We entered the following:

- Suffix DN: `dc=ibm,dc=com`

Note: This is the base suffix that will be used by the WebSphere Portal Server and the WebSphere Commerce Server.

- ▶ There is no space between the comma: `dc=ibm,dc=com`
- ▶ A convention used for naming the suffix DN is to use the TCP/IP domain/host name. For example, the IBM Directory Server host name is `wc1dap1.itso.ral.ibm.com`, and the suffix DN is `dc=ibm,dc=com`.

6. After adding the base DN suffix, stop and start the IBM Directory Server.

Import WebSphere Portal LDIF

To import the WebSphere Portal LDIF file, do the following:

1. We created a WebSphere Portal LDIF from the WebSphere Portal template LDIF. The ITSO created sample WebSphere Portal LDIF file can be found at `c:\sg246890-code\ldif\wp-itso.ldif`.

Note: For detailed information on obtaining and unpacking the ITSO provided sample code and configuration files, refer to 7.1, "ITSO sample code" on page 286.

2. Modify the `wp-itso.ldif` file for your environment. For example, the DN suffix `dc=ibm,dc=com` may need to be changed for your environment.
3. Stop the IBM Directory Server by selecting **Current State -> Start/Stop -> Stop**.

4. From the IBM SecureWay Directory Administration tool, select and expand **Database -> Import LDIF**.
5. Enter the full path of the LDIF file containing the WebSphere Portal data as follows and then click **Import**:
`c:\sg246890-code\ldif\wp-itso.ldif`
6. A summary report is displayed when the operation is finished.
7. After the LDIF file has been imported, delete the file from the system.

Note: Restarting the IBM Directory Server is not needed after importing an LDIF file.

Import WebSphere Commerce LDIF

To import the WebSphere Commerce LDIF file, do the following:

1. Ensure prerequisite software mentioned in the *Getting Started, IBM Commerce Enhancement Pack* have been completed.
2. We modified the WebSphere Commerce LDIF from the Commerce Enhancement Pack to include `cn=user` on the following line:

```
dn: uid=wcsadmin,cn=users,dc=ibm,dc=com
```

Without the update, the `wcsadmin` user was not visible from the DMT.

The ITSO created sample WebSphere Commerce LDIF file can be found:

```
c:\sg246890-code\ldif\wc-itso.ldif
```

Note: For detailed information on obtaining and unpacking the ITSO provided sample code and configuration files, refer to 7.1, "ITSO sample code" on page 286.

3. From the IBM Directory Server Administration tool, select and expand **Database -> Import LDIF**.
4. Click **Clear Results**.
5. Enter the full path of the LDIF file containing the WebSphere Commerce data as follows and then click **Import**:
`c:\sg246890-code\ldif\wc-itso.ldif`
6. A summary report is displayed when the operation is finished.
7. Click **Clear Results and restart server**.
8. After the LDIF file has been imported, delete the file from the system.

Note: Restarting the IBM Directory Server is not needed after importing an LDIF file.

DMT configuration for WebSphere Commerce

After the WebSphere Commerce LDIF file has been imported, we need to do the following using the DMT to configure the Directory Server:

- ▶ Start the DMT
- ▶ Change the wcsadmin password
- ▶ Create the wasadmin user
- ▶ Create an organization

Start the DMT

1. To start the Directory Management Tool, click **Start -> Programs -> IBM Directory Server 4.1 -> Directory Management Tool**.
2. By default, the DMT is started as an anonymous user. To log on as an authenticated user, do the following:
 - a. After starting the DMT, select and expand **Server -> Rebind**.
 - b. When the Rebind window appears, select **Authenticated**, enter the following and then click **OK**:
 - User DN: cn=root
 - User password: <password>

Change the wcsadmin password

To change the wcsadmin password, which was created via the wo-itso.ldif import, do the following:

Note: The wcsadmin password must match the password used when accessing WebSphere Commerce.

1. From the DMT, select and expand **Directory tree -> Browse tree**.
2. Select and expand the base DN (for example, dc=ibm,dc=com)
3. Double-click **uid=wcsadmin**.
4. Enter the wcsadmin password of the WebSphere Commerce node in the userPassword field of the DMT and then click **OK**.

Create the wasadmin user

To create the wasadmin user do the following:

Note: The wasadmin user will be used later as the Security Server ID during the configuration of WebSphere Security.

1. From the DMT, select and expand **Directory tree -> Browse tree**.
2. Select the base DN suffix (for example, dc=ibm,dc=com).
3. Click **Add** from the menu bar.
4. When the Add an LDAP Entry window appears, enter the following and then click **OK**:
 - Entry type: select **User**
 - Parent DN: dc=ibm,dc=com
 - Entry RDN: cn=wasadmin
5. A more detailed Edit an LDAP User window will appear. Enter the following and then click **Add**:
 - objectClass (Object class): select **Top** from the pull-down list
 - sn (Last name): wasadmin

We found that if we did not enter a value in the sn field, we received an error that would not allow the creation of this user.

From the Business tab, enter the following:

- userPassword: <wasadmin_password>

From the Other tab, enter the following:

- uid: wasadmin

Create an organization

Now we create the following organizations for the B2B store sample:

```
o=BuyerOrgA,dc=ibm,dc=com
o=BuyerOrgB,dc=ibm,dc=com
```

For each organization, do the following:

1. From the DMT, select and expand **Directory tree -> Browse tree**.
2. Select the base DN suffix (for example, dc=ibm,dc=com).
3. Click **Add** from the menu bar.
4. When the Add an LDAP Entry window appears, enter the following and then click **OK**:
 - Entry type: select **Organization**

- Parent DN: dc=ibm,dc=com
 - Entry RDN: o=BuyerOrgA
5. When the Add an LDAP Entry window appears, click **Add**.
 6. Repeat the process for BuyerOrgB.
 7. Click **Refresh tree**.
 8. After the refresh, the directory entries should look like Figure 5-4.

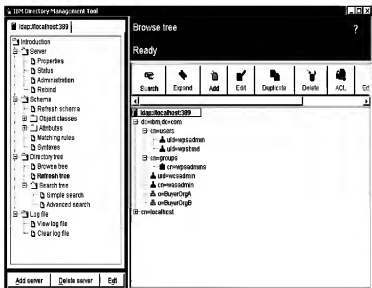


Figure 5-4 IBM Directory Server organization entries

9. Click **Exit** to close the DMT.

5.3.8 Configure WebSphere Commerce for LDAP

This section describes the configuration steps needed on the WebSphere Commerce node to configure WebSphere Commerce to use the IBM Directory Server (LDAP).

Note: More detailed information can be found in the following:

- ▶ *WebSphere Commerce V5.4 Handbook*, SG24-6567 redbook
- ▶ *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*

This section is organized as follows:

- ▶ Configure the WebSphere Commerce instance for LDAP
- ▶ Configure the `Idapentry.xml` mapping file
- ▶ Update WebSphere Commerce instance database for LDAP
- ▶ Verify the LDAP and WebSphere Commerce configuration

Configure the WebSphere Commerce instance for LDAP

To change the authentication mode used by WebSphere Commerce instance, complete the following steps on the WebSphere Commerce node:

1. Ensure the following tasks have been completed:
 - “Create a new suffix” on page 205
 - “Import WebSphere Commerce LDIF” on page 207
 - “DMT configuration for WebSphere Commerce” on page 208
2. Start the WebSphere Commerce Configuration Manager and log on.
3. From the left pane, select and expand **WebSphere Commerce** -> **<node> Instance List** -> **<instance_name>** -> **Instance Properties** -> **Member Subsystem**.
4. A pane should be displayed with the current authentication mode. From the Authentication Mode pull-down, select **LDAP**.
5. New fields should appear in the right pane under the Authentication Mode. For our example, we entered the following and then clicked **Apply**:
 - Authentication Mode: select **LDAP**
 - LDAP Version: **V3**
 - LDAP Type: select **IBM SecureWay** (predecessor to IBM Directory Server)
 - Single sign-on: check **Single Sign-on**
 - Host: `<ldap_hostname>`
For example, `wcldapl.itso.ral.ibm.com`
 - Port: **389**
 - Administrator Distinguished Name: `cn=root`
 - Administrator Password: `<password>`
 - Confirm Password: `<password>`
 - LDAP Authentication Mode: select **Simple**

- Time out: 240
 - Entry File Name: c:\ibm\wc\xml\ldap\ldapentry.xml
6. Close the Configuration Manager.
 7. An additional value needs to be modified manually in the WebSphere Commerce <instance>.xml file. The value cannot be changed from the Configuration Manager.
 - a. To change the value, open a text editor and edit the <instances>.xml file found at:


```
<WC_HOME>\instances\<instance_name>\xml\<instance>.xml
```
 - b. Locate the attribute MigrateUsersFromWCSDb in the <instance>.xml file. It is found inside the <Directory> tag in the <instance>.xml file, found inside <MemberSubSystem> tags.
 - c. Modify the attribute MigrateUsersFromWCSDb, set the value to ON (for example, MigrateUsersFromWCSDb="ON"). This property forces WebSphere Commerce to replicate the users from the WebSphere Commerce instance database to the LDAP directory database.
- Note:** The actual replication of information from LDAP to WebSphere Commerce does not occur until the WebSphere Commerce store registered user logs on to the store.
8. The changes made to the WebSphere Commerce <instance>.xml will not be in effect until the WebSphere Commerce <instance> application server has been restarted. We instruct you to restart the application server after modifying the ldapentry.xml file in the next step.

Configure the ldapentry.xml mapping file

For details on how to configure the ldapentry.xml mapping file, refer to the *WebSphere Commerce V5.4 Handbook*, SG24-6567. We will use an ldapentry.xml provided with the IBM Commerce Enhancement Pack - April 2003 Edition for commerce enabled portals (wcsportalldap.xml).

1. Change directory to the <WC_HOME>\xml\ldap.
2. Backup the existing ldapentry.xml file to ldapentry_org.xml.
3. Unzip the <CEP_HOME>\WebSphereCommerceEnabledPorta.zip.
4. Copy the Commerce Enhancement Pack provided in <CEP_HOME>\Base\WPS_WCS_LDAP_Integration_Config\wcsportalldap.xml to the <WC_HOME>\xml\Ndap\ldapentry.xml on the WebSphere Commerce node.

Note: The path and filename of the `ldapentry.xml` need to be the same as defined in the WebSphere Commerce `<instance>.xml` file. This information was entered using the Configuration Manager in "Configure the WebSphere Commerce instance for LDAP" on page 211.

5. Restart the WebSphere Commerce `<instance>` application server from the WebSphere Application Server Administrative Console.

Update WebSphere Commerce instance database for LDAP

The WebSphere Commerce instance database needs to be updated as part of the LDAP configuration for the organizations and users created in previous steps (review the contents of the `updatedbforldap.sql`).

1. Open a DB2 command window on the WebSphere Commerce node.
2. Connect to the WebSphere Commerce instance database.
`db2 connect to <wc_dbname> user <dbadmin> using <password>`
3. Change to the `<CEP_HOME>` unzip directory.
`<CEP_HOME>\Base\WPS_MCS_LDAP_Integration_Config`
4. Execute the following SQL script to update the WebSphere Commerce instance database:

```
db2 -tvf updatedbforldap.sql
```

Note: The `updatedbforldap.sql` file may need to be updated for your environment. For example, the SQL script include the Distinguished Name `dc=ibm,dc=com`.

5. Disconnect from the database.

```
db2 disconnect <wc_dbname>
```

Verify the LDAP and WebSphere Commerce configuration

After all of the configuration for WebSphere Commerce and LDAP has been completed, we recommend that you verify the functionality of the WebSphere Commerce tools before enabling WebSphere Application Server security, single sign-on, and SSL.

Refer to "Verify the administration tools" on page 205 for details.

5.3.9 Enable SSL between WebSphere Commerce and LDAP

By default, the connection between the WebSphere Commerce node and the Directory Server node is not secure. Within a test environment, this section is optional. Enabling SSL in a production environment is recommended.

Note: For details on how to SSL enable the connection between the nodes, refer to the following:

- ▶ *WebSphere Commerce V5.4 Handbook, SG24-6567 redbook*
- ▶ *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*

5.4 WebSphere Portal node implementation

This section describes the high-level steps to install the WebSphere Portal node within the ITSO runtime environment. This section includes the integration of WebSphere Portal V4.2.1 Enable with WebSphere Commerce V5.4.0.5 with the IBM Commerce Enhancement Pack - April 2003 Edition.

The WebSphere Portal node can be installed by one of the following approaches:

- ▶ WebSphere Portal Setup Manager umbrella install

The Setup Manager works reasonable well when all components are installed on the same node.

Note: For details on installing WebSphere Portal using Setup Manager, refer to the WebSphere Portal product documentation.

- ▶ WebSphere Portal component install

Installing the components of the WebSphere Portal environment is desirable in a multi-node environment such as our example, or if you are responsible for troubleshooting the environment. This type of install will provide a better understanding of how to verify and debug each component before proceeding to install and configure the next component of the WebSphere Portal node. This section documents the procedure for the component-based WebSphere Portal installation.

Note: For more information on the WebSphere Portal component installation, refer to the following:

- ▶ *WebSphere Portal V4.1, Windows 2000 Installation*, REDP3593 redpaper
- ▶ *IBM WebSphere Portal V4.1 Handbook*, SG24-6883 redbook

This high-level implementation steps for the WebSphere Portal node are as follows:

1. Windows 2000 Server installation
2. DB2 Server installation
3. IBM HTTP Server installation
4. WebSphere Application Server installation
5. WebSphere Portal installation
6. Commerce Enhancement Pack configuration

5.4.1 Windows 2000 Server installation

In preparation for the installation of DB2, ensure the following tasks have been completed:

1. Install Windows 2000 Server and Windows 2000 Service Pack 3.
2. Ensure an administrator user is logged in for installation of the WebSphere Commerce and supporting software with the following user rights:
 - Act as part of the operating
 - Create a token object
 - Increase quotas
 - Log on as a service
 - Replace a process level token
3. Install Internet Explorer 5.5 and service pack or higher.
4. Verify the configuration of the TCP/IP network (host name, IP address).

5.4.2 DB2 Server installation

In our example scenario, the DB2 Server is initially installed on the WebSphere Portal node to host the WebSphere Application Server repository for WebSphere Portal and WebSphere Personalization, and the WebSphere Portal databases. After the WebSphere Portal node has been verified to test the functionality with WebSphere Commerce, we will configure a connection to a remote DB2 Server and migrate the local application databases to the remote DB2 Server.

- ▶ Install DB2 UDB V7.2 Enterprise Edition

- ▶ Install DB2 V7 FixPak 7 (7.1.0.68)
- ▶ Update JDBC level to JDBC2

For details on the installation of the DB2 Server, refer to 5.7, "DB2 Server node implementation" on page 245.

5.4.3 IBM HTTP Server installation

For details on installing and configuring the IBM HTTP Server, refer to 5.2.3, "IBM HTTP Server installation" on page 176. The following tasks need to be completed:

- ▶ Install the IBM HTTP Server
- ▶ Configure the IBM HTTP Server
- ▶ Verify the IBM HTTP Server

Note: In our example, we first installed the IBM HTTP Server on the WebSphere Portal node. Later we migrated to a remote Web server configuration as described in 5.8, "Remote Web server node implementation" on page 250.

5.4.4 WebSphere Application Server installation

This section describes the high-level installation steps for the WebSphere Application Server on the WebSphere Portal node.

Note:

- ▶ For details, refer to the procedures documented in 5.2.4, "WebSphere Application Server installation" on page 177.
- ▶ The WebSphere Application Server repository database procedure is slightly different for this node.

1. Create the WebSphere Application Server repository database
 2. Install WebSphere Application Server V4.0.1
- Refer to "Install WebSphere Application Server V4.0.1" on page 178.

Note: During the installation of WebSphere Application Server you will be prompted for the WebSphere database name. We created this database in a previous step as was4wp. We did so to avoid a naming problem with other repositories in the event the database is migrated to a common remote database server.

3. Install WebSphere Application Server V4 FixPak 5 (V4.0.5)
Refer to "Install WebSphere Application Server V4 FixPak 5 (V4.0.5)" on page 179.
4. Install WebSphere Application Server V4.0.5 interim fixes
Refer to "Install WebSphere Application Server V4.0.5 interim fixes" on page 179.

Note: In addition to the interim fixes listed in "Install WebSphere Application Server V4.0.5 interim fixes" on page 179, install PQ66355 on the WebSphere Portal node.

The interim fix can be downloaded from:

<http://www.ibm.com/software/webservers/appserv/support/>

5. Configure the WebSphere Application Server
Refer to "Configure the WebSphere Application Server" on page 180.
6. Verify the WebSphere Application Server
Refer to "Verify the WebSphere Application Server" on page 180.

5.4.5 WebSphere Portal installation

This section describes how to install and configure the WebSphere Portal V4.2.1 server for the ITSO commerce enable portal environment.

Note: For detailed instructions on installing WebSphere Portal V4.2, refer to the following:

<http://publib.boulder.ibm.com/pwc/wp/42/smb/en/InfoCenter/index.html>

To install the WebSphere Portal V4.2.1, do the following:

1. Ensure the following Windows services are started:

WebSphere Portal node:

- DB2
- IBM HTTP Server
- IBM WS AdminServer 4.0

Directory Server node:

- DB2
- IBM HTTP Server
- IBM Directory Server V4.1

2. Insert the IBM WebSphere Portal Setup Manager CD in the CD-ROM drive. Navigate to the `wps` directory and run `install.bat`.
3. If a JDK is not installed, the Setup Manager will first install the JDK used by the installer and then continue.
4. Unless otherwise noted, we accepted the default options.
5. When the Installation Key window appears, we entered the values supplied in the Install Key Card PDF. Each version of WebSphere Portal for Multiplatforms (Enable, Extend, Experience) has a unique key.
6. We selected **Standard install** and then clicked **Next**.
7. We left the response file location field blank and clicked **Next**.
8. When the Component Selection window appears, we selected the following and then clicked **Next** (see Figure 5-5 on page 219):
 - Select **WebSphere Portal**. By selecting this option, many other components and subcomponents will be selected, which we accepted.

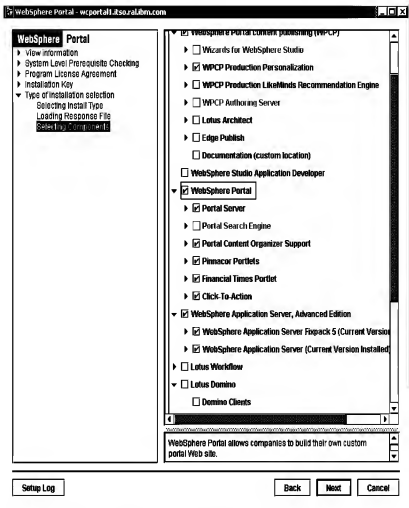


Figure 5-5 WebSphere Portal V4.2.1 components used for ITSO runtime environment

9. When asked if WebSphere Application Server Security is enabled, select **No** and then click **Next**.

In our scenario, we pre-installed WebSphere Application Server and did not enable security at this stage.

10. When prompted, we accepted the default settings for WebSphere Portal server and virtual host:

- Name of Application Server: WebSphere Portal
- Name of Virtual Host: default_host

Note: We recommend that you do not change these values.

11. When the Runtime Server Database window appears, we entered the following and clicked **Next**:

- Database Type: select **DB2 (local)**
- Database Name: PZNOB (accept default)
- User ID: <db2admin>
- Password: <db2admin_password>
- Confirm Password: <db2admin_password>
- Database Driver: COM.ibm.db2.jdbc.app.DB2Driver
- Driver Location: c:\ibm\sqllib\java\db2java.zip

12. When the Feedback Database window appears, we accepted the following default options and clicked **Next**:

- Database Type: select **DB2 (local)**
- Database Name: FdbkDB (accept default)
- User ID: <db2admin>
- Password: <db2admin_password>
- Confirm Password: <db2admin_password>
- Database Driver: COM.ibm.db2.jdbc.app.DB2Driver
- Driver Location: c:\ibm\sqllib\java\db2java.zip

13. When prompted to update to DB2 FixPak 8, select **Take No Action** and then click **Next**.

In our example, we are using DB2 FixPak 7, which is supported by WebSphere Commerce V5.4.0.5 and WebSphere Portal V4.2.1.

14. When asked whether WP CP (WebSphere Portal Content Publisher) should precompile JSPs, we selected **Yes** (the default) and clicked **Next**.

15. When prompted to select the Authentication mode for Portal Member Service, we selected **Database and LDAP Directory mode** (the default) and then clicked **Next**:

16. When prompted to configure security, we selected **Now** (the default) and then clicked **Next**.

17. When prompted for the LTPA password for WebSphere Application Server, we entered the following and then clicked **Next**:

- LTPA Password: <ltpa_password>
- Confirm Password: <ltpa_password>

18. When the Server Configuration window appears, we entered the following and then clicked **Next**:

- Install Directory: c:\ibm\PortalServer
- Host name for Web Server: wcportal1.itso.ral.ibm.com
- Base URI: /wps
- Home Page: /portal
- Customized Page: /myportal
- Proxy host: (leave blank)
- Proxy port: (leave blank)

19. When prompted to specify WebSphere Portal Administrator Settings, we accepted the default and clicked **Next** (wpsadmin).

20. When prompted for the LDAP Server type, we entered the following and then clicked **Next** (see Figure 5-6 on page 222):

- LDAP server type: select **IBM Directory Server**
- LDAP server: wcldapl.itso.ral.ibm.com
- User DN: cn=root
- User password: <password>
- Confirm password: <password>
- Suffix: dc=ibm,dc=com
- LDAP port number: 389
- Check **Make sure LDAP server is properly configured**.

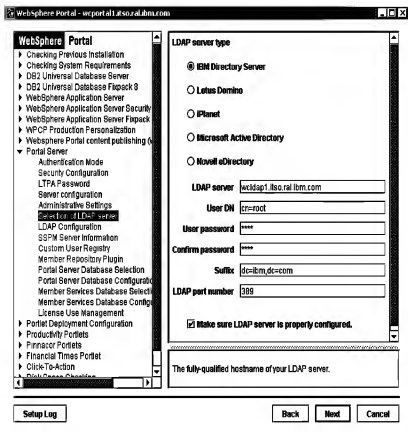


Figure 5-6 WebSphere Portal installation LDAP configuration settings

21. When prompted with the Distinguished Name entries for users and groups, we accepted the default settings and clicked **Next**.
22. When the Portal Server Database Selection window appears, we entered the following and then clicked **Next**:
 - Database back end: select **Local DB2 Universal Database™ Server**
 - Portal Server database configuration options: select **Create and initialize a new database (DB2 only)**
 - Do you want to share the database with Member Services?: select **Do not share the database**

23. When prompted to enter the DB2 information (Portal Server database), we entered the following and then clicked **Next**:
- Local Database User ID: db2admin
 - Local Database Password: <password>
 - Confirm Database Password: <password>
 - Local Database Alias Name: wpsdb
 - JDBC database driver: COM.ibm.db2.jdbc.DB2ConnectionPoolDataSource
 - JDBC URL prefix: jdbc:db2
 - JDBC driver library: c:\ibm\sqlib\java\db2java.zip
24. When prompted for the Member Services Database Selection, select **Create and initialize a new database (DB2 only)** and click **Next**.
25. When prompted to enter the Member Services Database Configuration, we entered the following and then clicked **Next**:
- Local Database User ID: db2admin
 - Local Database Password: <password>
 - Confirm Database Password: <password>
 - Local Database Alias Name: wmsdb
 - JDBC database driver: COM.ibm.db2.jdbc.DB2ConnectionPoolDataSource
 - JDBC URL prefix: jdbc:db2
 - JDBC driver library: c:\ibm\sqlib\java\db2java.zip
26. When prompted to configure License Use Management (LUM) for WebSphere Portal, select **Local License Server** (the default) and enter the LUM Server host name: wportal11.itso.ral.ibm.com. Then click **Next**.
27. The installer will detect that the Global Security Toolkit is installed (GSKit from IBM HTTP Server install). Click **Next**.
28. When prompted for the LDAP Directory, we entered c:\ibm\ldap and clicked **Next** to install the IBM Directory Server client.
- In our example, the IBM Directory Server V4.1 is installed on a separate node. This will install the IBM Directory Server client on the WebSphere Portal node.
29. When prompted for Pinnacor Portlets license agreement, select **Accept** and then click **Next**.
30. When prompted for Financial Times Portlets license agreement, select **Accept** and then click **Next**.
31. On the Click-To-Action Install Properties window, we entered the following and then clicked **Next**:
- WebSphere Application Server Installation Directory: c:\ibm\was
 - Portal Server Install Directory: c:\ibm\PortalServer

32. When prompted, click **Save Response** file for future installation use, and then click **Next**.
33. When the Installation options summary page appears, review settings and click **Next** to begin installation of files.
34. You will be prompted for the WebSphere Portal Family CD #5-2 (IBM Directory Server V4.1). In our example, this will install the IBM Directory Server client.
35. You will be prompted for the WebSphere Portal Family CD #4 (WebSphere Portal Content Publisher).
36. You will be prompted for the WebSphere Portal Family CD #7 (WebSphere Portal Server).
37. During the installation, an LDIF file will be generated that can be imported into the IBM Directory Server.

In our example, we provided a WebSphere Portal LDIF file and imported it as part of the Directory Server node configuration.
38. During the installation, you will be prompted to restart your Web server.

Note: The WebSphere Portal installation takes a long time to install. In our case, over an hour.

39. When complete, you will get a message "Installation Complete". Click **OK**, and then click **Finish**.

Congratulations, you have finished the WebSphere Portal installation.

WebSphere Portal install verification

After the installation of WebSphere Portal, you recommend that you do the following to verify the installation:

1. Start the WebSphere Application Server Administration Console. During the WebSphere Portal installation, WebSphere Security was enabled and the wpsbind user was created. Log on as wpsbind/wpsbind.
2. Enter the following URL to access the WebSphere Portal home page:
`http://<portal>/wps/portal`
3. Log on to the WebSphere Portal by clicking the login icon in the upper right of the home page. Log on as user wpsadmin and password wpsadmin.

5.4.6 Commerce Enhancement Pack configuration

This section describes the steps needed to install and configure the IBM Commerce Enhancement Pack - April 2003 Edition on the WebSphere Portal node in preparation for deploying commerce enabled portlets.

Note: For more information, refer to *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*.

To configure the Commerce Enhancement Pack on the WebSphere Portal node, do the following:

- ▶ Copy the Commerce Enhancement Pack APAR files
- ▶ Install the commerce enabled portal enterprise application

Copy the Commerce Enhancement Pack APAR files

This section describes how to copy the APAR level of the commerce enabled portal files over the original commerce enabled portal files.

1. Extract the WebSphereCommerceEnabledPortal.zip file provided with Commerce Enhancement Pack to C:\temp\CEP directory.
2. In 5.2.15, "Commerce Enabled Portal - APAR JR18068" on page 198, we installed the updated commerce enabled portal files to the WebSphere Commerce node. As per the instructions in the APAR readme, do the following:
 - a. Back up the Commerce Enabled Portal installation folder on the WebSphere Portal Server before proceeding. This folder was extracted from the WebSphereCommerceEnabledPortal.zip file, originally available from the Commerce Enhancement Pack (October 2002).
 - b. Go to the <WC_HOME>\service\efix\JR18068\install\WPS421 directory and copy those files to the directory where you unzipped the WebSphereCommerceEnabledPortal.zip file (for example, c:\temp\cep).
 - c. Copy the <WC_HOME>\service\efix\JR18068\CPSPZN\Installer\bin\cpspn.jar file to the c:\temp\cep\Base\PersonalizationUserHomePageBaseFolder\bin directory.
 - d. Copy the <WC_HOME>\service\efix\JR18068\CPSPZN\Installer\scripts\createCPSDB.sql file to the c:\temp\cep\commerceportal\Base\PersonalizationUserHomePageBaseFolder\scripts directory.

Install the commerce enabled portal enterprise application

To install the commerce enabled portal enterprise application on the WebSphere Portal node, do the following:

1. Copy the update c:\temp\cep directory (update commerce enabled portal) from the WebSphere Commerce node to the WebSphere Portal node.
2. Install the commerce enabled portal enterprise application.

- a. Change directory to the following:

```
C:\temp\CEP\Base\Personalization\UserHomePageBaseFolder\scripts
```

- b. Create the CPS database used by the Commerce Enhancement Pack by executing the following:

```
createCPSDB.bat
```

You will be prompted for the following:

- DB2 Admin UID: db2admin
- Host name: wcpportal1
- Password: <db2_admin_password>

- c. Ensure the WebSphere Administrative Server is running. This is required to create the CPS data source in the next step.
- d. Create CPS data source in WebSphere Application Server for the Commerce Enabled Portal Enterprise Application by executing the following:

```
createCPSDS.db2.bat
```

You will be prompted for the following:

- CPS database name: CPS
- Database admin UID: db2admin
- Database admin password: <db2_admin_password>
- DB2 JDBC Driver Path: c:\ibm\sqllib\java\db2java.zip
- WebSphere Application Server node name: wcpportal1
- WebSphere Application Server path: c:\ibm\was

Note: The WebSphere node name is case sensitive.

Since the WebSphere Portal node is now configured for LDAP, a login panel will be displayed. We logged on as wpsbind.

- e. Install the Commerce Portal Server enterprise application in the WebSphere Application Server.

```
createCPSEA.bat
```

You will be prompted with the following:

- WebSphere Application Server node name: `wcportal1`
- Install folder: `c:\ibm\cps`
- WebSphere Application Server path: `c:\ibm\was`

Note: WebSphere node name is case sensitive.

You will get a login window, since WebSphere Application Server security has been enabled. Login with `wpsbind`.

- f. Copy the JAR files to WebSphere Application Server by running the following command:

```
modifyWAS.bat
```

You will be prompted with the following:

- WebSphere Application Server Path: `c:\ibm\was`

Note: You will get a login window, since WebSphere Application Server security has been enabled. Login with `wpsbind`.

3. For personalized accounts for the default store, run the following (optional):

```
installNewWPSAttributes.bat
```

You will be prompted with the following:

- WebSphere Application Server Path: `c:\ibm\was`
- Portal Path: `c:\ibm\PortalServer`

4. Verify that WebSphere Portal is functioning properly:

- a. Ensure the following are started:

- DB2 - Windows services
- IBM HTTP Server - Windows services
- IBM WS AdminServer 4.0 - Windows services
- WebSphere Portal - application server (restart)

- b. Enter the following WebSphere Portal home page URL in a Web browser:

```
http://wcportal1.itso.ral.ibm.com/wps/portal
```

- c. Verify that you can log in as `wpsadmin`.

- d. Verify that you can log in with a user ID registered in WebSphere Commerce such as `wcsadmin`.

5.5 Enabling single sign-on between WebSphere Portal and WebSphere Commerce

Now that the WebSphere Commerce node, Directory Server node, and WebSphere Portal node have been installed and configured, we can enable WebSphere security and enable single sign-on (SSO). This section explains how to enable single sign-on (SSO) of WebSphere Commerce with other secure applications such as WebSphere Portal, using the same Directory Server (LDAP). SSO allows the user to move between different applications (servers) without being prompted for a user ID and password (or certificate).

This section is organized as follows:

1. Overview on single sign-on (SSO)
2. WebSphere Portal configuration for SSO
3. WebSphere Commerce configuration for SSO
4. Verify single sign-on (SSO) configuration

Note: For more information on WebSphere Security and how to SSL enable the connection between the nodes, refer to the following:

- ▶ *WebSphere Commerce V5.4 Handbook*, SG24-6567 redbook
- ▶ *IBM WebSphere V4.0 Advanced Edition Handbook*, SG24-6176 redbook
- ▶ *IBM WebSphere V4.0 Advanced Edition Security*, SG24-6520 redbook
- ▶ *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition* product guide

5.5.1 Overview on single sign-on (SSO)

To enable SSO between different servers, the Lightweight Third Party Authentication (LTPA) mechanism is used. This mechanism uses a flag called an *LtpaToken*, which contains the user authentication information, the network domain in which the SSO should be valid, and the expiration time after the user is required to re-authenticate. The *LtpaToken* is encrypted using LTPA keys shared for all the SSO participating servers.

The token is created when the user successfully authenticates the first participating application (server). The server sends a *transient* cookie to the Web browser client. This means that the cookie resides in the Web browser memory and is not stored on the user's computer system. This type of cookie expires when the user closes the browser and is easily recognized by its name, *LtpaToken*.

The requirements for enabling SSO are as follows:

- ▶ Use the same LDAP directory for authentication.
- ▶ All SSO participating servers must be in the same DNS domain.
- ▶ The URLs must include the DNS domain (no IP addresses or host names).
- ▶ The Web browsers must be configured to accept cookies.
- ▶ The servers' time and time zone must be synchronized (SSO token expiration time is absolute).
- ▶ All servers must share the LTPA keys to generate and decode the LtpaTokens.

WebSphere Commerce SSO implementation

The following considerations need to be kept in mind when planning to enable SSO with WebSphere Commerce:

- ▶ WebSphere Commerce cannot create a LtpaToken. The WebSphere Commerce Server does not create an LtpaToken when a user is authenticated using the default member subsystem login pages. A simple solution to this is to create a small WebSphere application just to perform the authentication. A production runtime environment may have something like a portal server with directory services that creates the LtpaTokens. The following steps will explain how to use the WebSphere security services to configure an existing application.
- ▶ To automatically sign on to the WebSphere Commerce Administrative tools, you must connect directly to the URL of one of the pages displayed after performing the logon. If you point to the logon page, a user ID and password must be supplied, because the page does not check if the LtpaToken is received.

Scenario for single sign-on

The following sections show how to implement a real SSO scenario. The scenario consists of two WebSphere Application Server applications on two different nodes:

- ▶ Node A running WebSphere Portal V4.2.1
- ▶ Node B running WebSphere Commerce V5.4.0.5

The scenario describes how to set up two different nodes for SSO and how a user logged into Node A can access the Node B application without being prompted for a user ID and password.

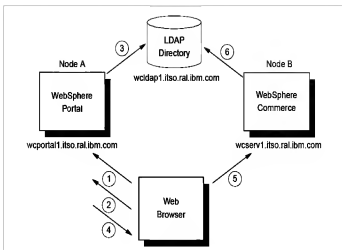


Figure 5-7 Sample scenario for SSO

The SSO authentication works as follows (see Figure 5-7):

1. The user enters the URL for the sample application on Node A. Since the application is configured to use LDAP as an authentication method, the WebSphere Application Server redirects to a logon page. In the case of WebSphere Portal, we will use the logon provided with the product.
2. The users enter a user ID and password.
3. The WebSphere Portal authenticates the users against the LDAP directory server.
4. The authentication is successful, so the WebSphere Application Server executes the application and the response is returned to the user. As SSO is also enabled, the WebSphere Application Server creates an LtpaToken and sends a cookie to the Web browser client.
5. The user accesses the WebSphere Commerce store on Node B. As Node B belongs to the same DNS domain, the Web browser client sends the cookie received in the previous step from Node A to Node B.
6. The WebSphere Commerce Server reads the cookie and decrypts the Distinguished Name (DN) of the user logged in and verifies the DN exists in the LDAP directory. If the DN exists, the user is logged into the WebSphere Commerce store.

5.5.2 WebSphere Portal configuration for SSO

This section describes how to configure the WebSphere Portal node for single sign-on use with the WebSphere Commerce node.

The section is organized as follows:

- ▶ Enable WebSphere security/SSO for WebSphere Portal node
- ▶ Verify the WebSphere security configuration
- ▶ Verify the WebSphere Portal
- ▶ Extract certificate from WebSphere Commerce Web server
- ▶ Certificate trust security configuration

Enable WebSphere security/SSO for WebSphere Portal node

In our example, WebSphere Commerce and WebSphere Portal are on separate nodes. For this reason, we must export the LTPA key (LtpaToken) from the WebSphere Portal node from WebSphere security, and import the key on the WebSphere Commerce node.

To configure global security settings for SSO in WebSphere, complete the following steps:

1. Open the WebSphere Administrative Console on the WebSphere Portal node. Log on as `wasbind`, which is the default WebSphere Security Server ID set by the WebSphere Portal installation.
2. From the WebSphere Administrative Console menu bar, select **Console -> Security Center**.
3. Click the **General** tab and check **Enable security**.
4. Click the **Authentication** tab and choose **Lightweight Third Party Authentication (LTPA)**.
5. New options for LTPA setting will be available on the window. Specify the following LTPA settings:

Note: Many of the options listed will have the proper value already set by the WebSphere Portal install. We recommend that you carefully review each setting.

- Token Expiration: 240

How many minutes can pass before a client using an LtpaToken must authenticate again in the Token Expiration field.

- Ensure the **Enable Single Sign-On (SSO)** is checked. The Domain field will be enabled once Enable Single Sign-On is checked.

- Domain: itso.ral.ibm.com

Enter a DNS domain name in the Domain field. In our example we set this domain to itso.ral.ibm.com. This domain name is used when the HTTP cookie is created for SSO and determines the scope to which SSO applies.

- From the same page, select the **LDAP** radio button and enter the following for accessing the LDAP directory:
 - Security Server ID: uid=wpsbind,cn=users,dc=ibm,dc=com

Note: The Security Server ID is set during the WebSphere Portal installation. In our example, it was set to uid=wpsbind,cn=users,dc=ibm,dc=com.

We found that by changing the Security Server ID, we were not able to log on to WebSphere Portal with the wpsadmin or wpsbind user IDs.

- Security Server Password: <password>
- Host: <fully_qualified_hostname_of_LDAP_server>
For example, we entered wcldap1.itso.ral.ibm.com.
- Directory Type: Select **Custom**
- Port: 389
- Base Distinguished Name: dc=ibm,dc=com
- Base entry DN where the users can be found.
- Bind Distinguished Name: (we left field blank on the WebSphere Portal node, which is the default from the WebSphere Portal installation)
- This is the IBM Directory Server user that will do the bind to the LDAP directory (used to rebind as an authenticated user in the DMT).
- Bind Password: (left field blank)

After entering these values, the window should look like Figure 5-8 on page 233. Do not click Apply or OK just yet.

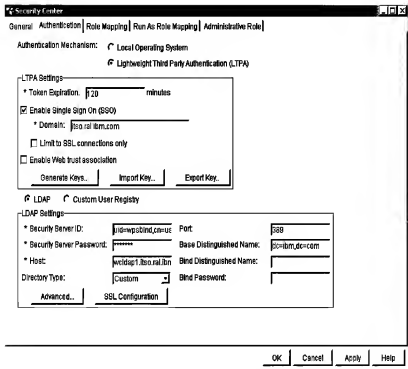


Figure 5-8 WebSphere security SSO and LDAP settings for the WebSphere Portal node

6. From the same page, click the **Generate Keys...** button to create the LTPA keys for encrypting the LtpaToken.
 - a. You will be prompted for an LTPA password to protect the set of encryption keys. Type in your LTPA password (specified during WebSphere Portal installation).
 - b. The LTPA keys will be shared with the WebSphere Commerce Server. Click **OK**.

Note: A message Command "Generate LTPA keys" created successfully should appear in the message log of the WebSphere Administrative Console.

7. Once the LTPA keys are generated, click **Export Key...** You will be prompted with an **Export To File** window. For example, we created an LTPA key file called `wp_wasltpa.key` in the `c:\ibm\was\bin` directory.
Keep this file in a secure place, because these keys will be imported into the WebSphere Application Server of the WebSphere Commerce node.
8. Now that all Security and single sign-on settings have been entered, click **OK**
9. Verify that the settings were saved by reviewing the file:
`<WAS_HOME>\properties\sas.server.props`
10. The WebSphere Administrative Server will need to be restarted for the changes to take effect. If no values are changed, you do not need to restart.

Verify the WebSphere security configuration

To verify that WebSphere security is configured correctly, do the following:

1. Start the Default Server application server from the Administrative Console.
2. Enter the following URL in a Web browser:
`http://<wcpportal_hostname>/webapp/examples/showCfg`
3. The WebSphere basic authorization page should appear. We entered the following and then clicked **Submit Login**:
 - Account: `wpsbind`
 - Password: `<password>`You should see the IBM WebSphere Web Container Configuration.
4. Stop the Default Server application server.

Verify the WebSphere Portal

Verify that the WebSphere Portal is functioning properly after the WebSphere security configuration as follows:

1. Ensure the following Windows services are started:
 - DB2 - Windows services
 - IBM HTTP Server - Windows services
 - IBM WS AdminServer 4.0 - Windows services
2. Start the WebSphere Portal application server.
 - a. Start the WebSphere Administrative Console.
 - b. When prompted for the logon information, we entered the following and then clicked **OK**:
 - Real/Cell Name: `wcdap1.itso.ral.ibm.com®` (this field is greyed out, cannot enter info).
 - User Identity: `wpsadmin`

- User Password: <password>
- c. Select and expand **WebSphere Administrative Domain** -> **Nodes** -> **<node>** -> **Application Servers**.
 - d. Select the **WebSphere Portal** application server.
 - e. Right-click **Start**.
 3. Enter the following WebSphere Portal home page URL in a Web browser:
http://wcportal1.itso.ral.ibm.com/wps/portal
 4. Verify that you can log in as wpsadmin.
 5. Verify that you can log in with a user ID registered in WebSphere Commerce such as wcsadmin.

Extract certificate from WebSphere Commerce Web server

On the WebSphere Commerce node, extract the certificate from the IBM HTTP Server as follows:

1. Start the IBM HTTP Server IKeyMan Utility by clicking **Start** -> **Programs** -> **IBM HTTP Server** -> **Start Key Management Utility**.
2. From the menu bar, click **Key Database File** -> **Open**. Enter the path to the keyfile (key store). For example, c:\ibm\http\ssl\http_key.kdb.
The key file is referenced in the <HTTP_HOME>\conf\httpd.conf file.
3. You will be prompted to enter the key store database password.
4. Select the certificate (for example, http_ssl) and click **Extract Certificate**.
5. When the Extract Certificate window appears, enter the following and click **OK**:
 - Data type: select **Base64 encoded ASCII** data
 - Certificate file name: wc_http_cert.arm
 - Location: c:\ibm\http\ssl
6. Close the IBM Key Management Utility.

Certificate trust security configuration

This section describe how to configure the IBM Secure Sockets Extension for WebSphere security and how to trust certificates between the WebSphere Portal node and WebSphere Commerce node.

1. Enable the IBM Secure Sockets Extension API by editing the security properties file on the WebSphere Portal node:
<WAS_HOME>\java\jre\lib\security\java.security

Verify that the following line exists and add it if necessary:

```
security.provider.x=com.ibm.jsse.JSSEProvider
```

Where *x* is the sequence number (for example, 2 or 3).

2. Restart WebSphere Application Server Administrative Server (not needed if java.security was not modified) on the WebSphere Portal node.
3. Import the IBM HTTP Server certificate from WebSphere Commerce node to the WebSphere Portal Server.
 - a. Copy the `wc_http_cert.arm` file from the WebSphere Commerce node to the `<WAS_HOME>/java/jre/lib/security` directory on the WebSphere Portal node.
 - b. Issue the following command from `<WAS_HOME>/java/jre/lib/security` directory:

```
..\..\bin\keytool -import -alias MCCERT -file wc_http_cert.arm -keystore cacerts
```
 - c. This command will import the SSL certificate into the default *cacerts* keystore. The default password for the *cacerts* keystore is *changeit*. Alias identifies this certificate in the keystore and it can be any name you choose.
 - d. You will be prompted to trust the certificate (self-signed). Enter Yes.

5.5.3 WebSphere Commerce configuration for SSO

To configure WebSphere Application Server on the WebSphere Commerce node and change the WebSphere Commerce instance configuration to support SSO, do the following:

- ▶ Enable WebSphere security/SSL for WebSphere Commerce
- ▶ Modify WebSphere Commerce instance configuration
- ▶ Modify `ldapentry.xml` file
- ▶ Modify existing users for objectClass `ePerson` (optional)

Enable WebSphere security/SSL for WebSphere Commerce

To enable WebSphere security and SSO on the WebSphere Commerce node, do the following:

1. Copy the exported LTPA key file from the WebSphere Portal node to the WebSphere Commerce node.

For example, we copied the `c:\ibm\was\bin\wp_wasltpa.key` from the WebSphere Portal node to the `c:\ibm\was\bin` directory on the WebSphere Commerce node.

2. Start the WebSphere Administrative Console on the WebSphere Commerce node.
3. From the WebSphere Administrative Console menu bar, select **Console -> Security Center**.
4. Click the **General** tab and check **Enable security**.
5. Click the **Authentication** tab and choose **Lightweight Third Party Authentication (LTPA)**.
6. New options for LTPA setting will be available on the window. Specify the following LTPA settings:

Note: Many of the options listed will have the proper value already set. We recommend that you carefully review each setting.

- Token Expiration: 240

This sets how many minutes can pass before a client using an LtpaToken must authenticate again in the Token Expiration field.

- Ensure the **Enable Single Sign-On (SSO)** is checked. The Domain field will be enabled once Enable Single Sign-On is checked.

- Domain: itso.ra1.ibm.com

Enter a DNS domain name in the Domain field. In our example we set this domain to itso.ra1.ibm.com. This domain name is used when the HTTP cookie is created for SSO and determines the scope to which SSO applies.

From the same page, select the **LDAP** radio button and enter the following for accessing the LDAP directory:

- Security Server ID: wasadmin
- Security Server Password: <password>
- Host: <fully_qualified_hostname_of_LDAP_server>
For example, we entered wcldap1.itso.ra1.ibm.com.
- Directory Type: Select **Custom**
- Port: 389
- Base Distinguished Name: dc=ibm,dc=com
Base entry DN where the users can be found.
- Bind Distinguished Name: cn=root

This is the IBM Directory Server user that will do the bind to the LDAP directory (used to rebind as an authenticated user in the DMT).

– Bind Password: <password>

After entering these values, the window should look like Figure 5-9. Do *not* click Apply or OK just yet.

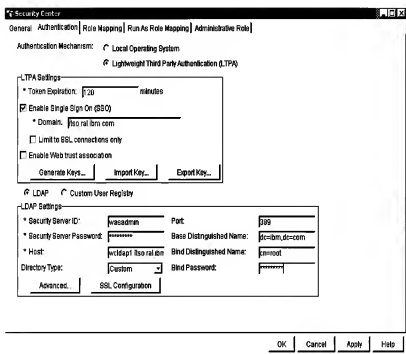


Figure 5-9 WebSphere security SSO and LDAP settings for the WebSphere Commerce node

7. Click the **Advanced** tab, enter the following and then click OK (see Figure 5-10 on page 239):
 - Initial JNDI Context Factory: `com.sun.jndi.ldap.LdapCtxFactory` (the default)
 - User Filter: `(&(uid=%v)(objectclass=inetOrgPerson))`
 - Group Filter: `(&(cn=%v)(objectclass=groupOfUniqueNames))`
 - User ID Map: `*:uid`
 - Group ID Map: `*:cn`
 - Group Member ID Map: `groupOfUniqueNames:uniqueMember`
 - Certificate Mapping: `Exact Distinguished Name`

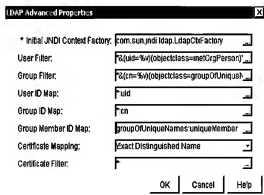


Figure 5-10 LDAP Advanced Properties

8. From the same page, click **Import Key...**
 - Select the Key file copied from the WebSphere Commerce node. In our example, we imported the c:\ibm\was\bin\wp_wasitpa.key copied above from WebSphere Portal node to the WebSphere Commerce node.
 - Type the same password entered when the key was generated. Re-enter the password and click **OK**.
9. Click **OK** to accept the security settings.
10. Restart the WebSphere Administrative Server for the security changes to take effect before proceeding (adminserver.bat).
11. Start the WebSphere Administrative Console on the WebSphere Commerce node. Log on using the Security Server ID wpsbind.
12. From the WebSphere Administrative Console menu bar, select **Console -> Security Center**.
13. Click the **Role Mapping** tab.
 - a. Select the enterprise application for the WebSphere Commerce instance (for example, WebSphere Commerce Enterprise Application - wc1).
 - b. Click **Edit Mappings....**
 - c. Under Roles, select **WCSecurityRole**, and then click **Select**.
 - d. Check the **Select users/groups** check box.
 - e. Enter wpsbind in the search field, and then click **Search**.

- i. Select the **cn=wasadmin,dc=ibm,dc=com** under Available Users/Groups, and then click **Add**. The result of this action should look like Figure 5-11.

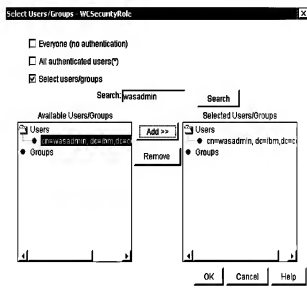


Figure 5-11 Select Users/Groups - WLSecurityRole

- g. Click **OK** on the next two pop-up windows.
- h. Click **Apply** and then **OK**.

Modify WebSphere Commerce instance configuration

Configure the WebSphere Commerce instance to use WebSphere Application Server security and enable SSO by completing the following steps:

1. We will need to temporarily start the WebSphere Administrative Server from Windows services so that the WC Configuration Manager recognizes the Administrative Server is started. Ensure you have stopped the server if you are running `adminserver.bat`.
2. Start the WC Configuration Manager Server Windows services.
3. Start the WebSphere Commerce Configuration Manager. Enter the user ID `wbadmin` and `<password>` when prompted.

Note: Remember to start the IBM WC Configuration Manager Server Windows services prior to launching the application.

4. Select **WebSphere Commerce** -> <node-name> -> Instance List -> <instance_name> -> **Instance Properties** -> **Security**.
5. Check the **Enable Security** check box.
6. A message window advises you that WebSphere Application Server security has to be set up before enabling security. Click **Yes**, assuming you have configured this option by following this procedure.
7. Select **LDAP User Registry** and enter the user ID and password of the WebSphere Security Server ID (for example, wpsbind) and then click **Apply**.
8. When the message Successfully configured security for WebSphere Commerce appears, click **OK**.
9. Select **WebSphere Commerce** -> <node-name> -> Instance List -> <instance_name> -> **Instance Properties** -> **MemberSystem**. In the right pane, check the **Single sign-on** check box and click **Apply** (should already be set).
10. Close the WebSphere Commerce Configuration Manager.
11. Modify the <wc_instance>.xml file.
 - a. Using a text editor, open the WebSphere Commerce instance configuration file <instance_name>.xml located in the <wc_home>/instances/<instance_name>/xml directory.
 - b. Locate the tag <SessionManagement>.
 - c. Under the tag <cookie > add the attribute sslauth="false".

Note: It is our understanding that this value is used to accept the LtpaToken property when it comes from a non-SSL page and vice versa.

The resulting code should look like Example 5-2.

Example 5-2 Sample <wc_instance>.xml for security

```
<SessionManagement name="Session Management">
  <url-rewriting display="false"
    enabled="false" />
  <cookie acceptance="false"
    age="-1"
    display="false"
    domain=""
```

```
        enabled="true"  
        path="/" *  
        sslauth="false"  
        persistence="wcs" />  
</SessionManagement>
```

- d. Save the file and close the text editor.

Modify ldapentry.xml file

To implement SSO, the entries for users in the LDAP directory must include the objectclass ePerson. In the default mapping file (ldapentry.xml), the entries created by WebSphere Commerce member subsystem do not include the ePerson objectclass.

To add the objectclass ePerson, complete the following steps:

1. Open the file ldapentry.xml with a text editor.

The ldapentry.xml file should be located in <wc_home>/xml/ldap/.

2. Locate the tag <ldapocs> and add the objectclass ePerson.

```
<ldapocs objClass="top;person;organizationalPerson;inetOrgPerson;ePerson"/>
```

Important: Only add the ePerson objectclass for the entryType="User" (do not add to Organization or OrganizationalUnit).

3. Save the changes and close the text editor.
4. Restart the WebSphere Commerce <instance> application server.

Modify existing users for objectClass ePerson (optional)

From this moment on, all new users registered will be created including the objectclass ePerson, but all users already registered in need of SSO on the WebSphere Commerce node will need to be updated in the LDAP directory.

Note: For our example, this is not necessary. The only users we have created that do not have the objectClass ePerson are wpsadmin and wpsbind, neither of which require single sign-on with the WebSphere Commerce node.

To modify the existing users, complete the following steps on the Directory Server node:

1. Before proceeding, we recommend that you back up the IBM Directory Server database (for example, LDAPDB2) containing directory information. Refer to "Back up a DB2 database" on page 375.
2. Open the IBM Directory Server Web Administration tool.

3. Log on as `cn=root`.
4. Select **Database -> Export LDIF**.
5. Specify a file name and export all the entries (we accepted the default `c:\swd_home>\var\ldap\export.ldif`). Click **Export**.
6. Using a text editor, open the exported LDIF file and add the following line to all the User entries:

```
objectclass: ePerson
```
7. Save the changes and close the text editor.
8. Using the DMT, rebind and connect as an authenticated user, `cn=root`.
9. Back up the LDAP database LDAPDB2. Refer to "Back up a DB2 database" on page 375 for details.
10. Select **Browse tree** and delete all the entries under the DN that are going to be updated.
11. Use the IBM SecureWay Directory Web Administration tool to import the modified entries. Select **Database -> Import LDIF** and type the name of the modified LDIF file (for example, `c:\swd_home>\var\ldap\export.ldif`) and then click **Import**.

You may have to click **Clear Results** prior to the import if not displayed.

Depending on how many users you have updated/deleted, you will get an error message saying that the update was not successful, but it will tell you how many entries were updated.
12. Refresh the tree in DMT and verify that the new entries now include the `ePerson` objectclass. Expand the tree and double-click one of the modified entries. Expand the list of values for the objectclass (Object Class) field. The `ePerson` value should now be included in the list.

Is this the best way to update the object class?

An additional class can be added using the DMT. The problem is you have to do it one by one, and it is probably better to use LDIF files for large amounts of entries to update.

5.5.4 Verify single sign-on (SSO) configuration

To test the implementation, we must connect to the sample application, authenticate, and then move to WebSphere Commerce without being prompted for a password.

To test the SSO enablement, complete the following steps:

1. Ensure the following are started on each of the nodes:
 - WebSphere Portal node:
 - DB2
 - IBM HTTP Server
 - IBM WS AdminServer 4.0
 - WebSphere Portal application server
 - Directory Server node:
 - DB2
 - IBM HTTP Server
 - IBM Directory Server V4.1
 - WebSphere Commerce node:
 - DB2
 - IBM HTTP Server
 - WebSphere Administrative Server (adminserver.bat)
 - WebSphere Commerce Payments application server and IBMPayServer.cmd
 - WebSphere Commerce <instance> application server
2. Close all Web browser windows opened. Launch a new Web browser window and type the URL for the WebSphere Portal:
`http://wcpportal1.ftso.ra1.ibm.com/wps/portal`
3. Log on to the WebSphere Portal with the wcsadmin user (has ePerson objectClass), and then click **Submit Login**.

Important: The selected user entry in the LDAP directory must have been updated to include the objectClass ePerson as described in, "Modify existing users for objectClass ePerson (optional)" on page 242.

4. On the WebSphere Commerce node, start the Default Server.
5. In the same Web browser window, enter the URL for showCfg sample application on the WebSphere Commerce node:

`http://<wc_hostname>/webapp/examples/showCfg`

Note: If you have already published a WebSphere Commerce store, you can enter the store URL. For example:

`http://<wc_hostname>/webapp/wcs/stores/servlet/tooltech/index.jsp`

Congratulations! You have now configured the WebSphere Commerce node, WebSphere Portal node, and Directory Server node for single sign-on.

5.6 Deploying the ITSO B2B CEP store

Refer to 9.1, "Deploy the ITSO B2B CEP store to the runtime environment" on page 310 for details on deploying the ITSO B2B CEP store.

5.7 DB2 Server node implementation

This section describes the high-level steps to install the DB2 Database Server node within the ITSO test environment in preparation for hosting databases for the WebSphere Application Server repository, WebSphere Commerce instance database, WebSphere Commerce Payments database, and WebSphere Portal database.

Note: More detailed information can be found in the *WebSphere Commerce V5.4 Handbook*, SG24-6567 redbook and the product installation guides.

The high-level installation and configuration procedures for the DB2 Server node are as follows:

- ▶ Windows 2000 Server installation
- ▶ DB2 Server installation
- ▶ Verify the DB2 instance TCP/IP connection port
- ▶ Migrate WebSphere Commerce databases to remote DB2 Server

5.7.1 Windows 2000 Server installation

In preparation for the installation of WebSphere Commerce and supporting components, ensure the following tasks have been completed:

1. Install Windows 2000 Server and Windows 2000 Service Pack 3.
2. Ensure an administrator user is logged in for installation of the WebSphere Commerce and supporting software with the following user rights:
 - Act as part of the operating
 - Create a token object
 - Increase quotas
 - Log on as a service
 - Replace a process level token
3. Install Internet Explorer 5.5 and service pack or higher.

4. Verify the configuration of the TCP/IP network (host name, IP address).

5.7.2 DB2 Server installation

This section highlights the key steps for installing and configuring the DB2 Server for use with WebSphere Application Server and WebSphere Commerce.

The DB2 Server installation is organized as follows:

- ▶ Install DB2 UDB V7.2 Enterprise Edition
- ▶ Install DB2 V7 FixPak 7 (7.1.0.68)
- ▶ Update JDBC level to JDBC2

Note: For detailed installation instructions, refer to the following:

- ▶ *WebSphere Commerce V5.4 Handbook*, SG24-6567 redbook
- ▶ *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Windows* product guide

Install DB2 UDB V7.2 Enterprise Edition

The high-level steps to install the IBM DB2 UDB V7.2, Enterprise Edition, are as follows:

1. Insert the IBM DB2 UDB V7.2, Enterprise Edition CD and run **Setup**.
2. We accepted the default options unless noted as follows:
 - Select **DB2 Enterprise Edition** and **DB2 Application Development Client**.
 - Select **Custom** for the installation type. Refer to the noted documentation for details.
 - We installed DB2 to the c:\ibm\sqlib directory.
 - Create an instance.

Install DB2 V7 FixPak 7 (7.1.0.68)

We installed DB2 V7 FixPak 7 (7.1.0.68), which can be downloaded at:

ftp://ftp.software.ibm.com/ps/products/db2/fixes/english-us/db2ntv7/FP7_WR21311/

Update JDBC level to JDBC2

Update the JDBC level to JDBC2 as follows:

1. Stop all DB2 Windows services.
2. Run `usejdbc2.bat` found in the <DB2_HOME>\java12 directory.

3. The inuse file found in <DB2_HOME>\java12 directory should state the following:

JDBC 2.0

4. To verify the JDBC functionality (optional).

We have included an IBM site where a JDBC test tool and instructions (jdbctest.java, jsread2.html) for verification can be downloaded and run on your system. Enter the following FTP link in a Web browser:

<ftp://ftp.software.ibm.com/software/websphere/info/tools/jdbctest>

5.7.3 Verify the DB2 instance TCP/IP connection port

To verify the DB2 instance TCP/IP connection port, do the following:

1. On the DB2 Server node, execute the following command in DB2 command Window:

```
db2 get dbm cfg
```

2. In the output of the command, search for the following, as seen in Figure 5-12 on page 248:

```
TCP/IP Service Name (SVCENAME) =
```

This is the DB2 instance connection port. If instead of a number you see a name, such as db2cDB2, look up the corresponding port number for this name in the C:\WINNT\system32\drivers\etc\services file. (On AIX, Solaris and Linux, this file is called /etc/services.) This port number or service name will be needed during the DB2 client configuration for Commerce Application Server node (WebSphere Application Server, WebSphere Commerce, WebSphere Commerce Payments) and the WebSphere Portal node.



Figure 5-12 DB2 instance TCP/IP connection port

5.7.4 Migrate WebSphere Commerce databases to remote DB2 Server

In our example, we installed the DB2 Server on the Commerce Application Server node. Now that we have verified that the node is working, we can move the application databases to the remote DB2 Server. This step is optional.

Note: There are many ways to configure this type of environment. It is possible that the DB2 client and server be configured from the start. We chose this approach to demonstrate how to migrate databases and to avoid problems with the IBM Commerce Enhancement Pack - April 2003 Edition installer when using a remote database server node.

The DB2 Server includes the components of the DB2 client. For this reason, we do not have to install the DB2 client. The procedure documented will explain how to migrate application databases (WebSphere Application Server, WebSphere Commerce and WebSphere Commerce Payments databases) and configure the DB2 client on the Commerce Application Server to communicate with the remote DB2 database server.

Note: For detailed installation instructions, refer to the following:

- ▶ *WebSphere Commerce V5.4 Handbook*, SG24-6567 redbook
- ▶ *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Windows* product guide

Back up the application databases

For information on backing up DB2 databases, refer to “Back up a DB2 database” on page 375.

Restore the application databases

For information on restoring DB2 databases, refer to “Restore a DB2 database” on page 376.

Configure and verify the DB2 client/server connectivity

This section describes how to configure the DB2 client and server and verify that they are communicating properly.

Catalog the TCP/IP node

From a DB2 command window, type the following command:

Syntax:

```
db2 catalog tcpip node <node_name> remote <server_name> server  
<port_number>
```

The <port_number> is the DB2 instance connection port found on the server in the services file. Alternatively, in place of the port number a service name can be used. If a service name is used, the port and service name must be added to the DB2 client system services file so that it can resolve where to find the system.

Example:

```
db2 catalog tcpip node wcdw2 remote wcdw2 server 50000
```

Attach to the remote DB2 Server

From a DB2 command window, type the following command:

Syntax:

```
db2 attach to <node_name> user <db2_username> using <db2_user_password>
```

Example:

```
db2 attach to wcdw2 user db2admin using <password>
```

Catalog the databases

Once you have attached to the remote DB2 Server, catalog the databases from a DB2 command window:

```
db2 catalog db <db_name> at node <node_name>
```

Note: When migrating the databases from the Commerce Application Server node to the DB2 Database Server node, you will need to catalog the following databases:

- ▶ WebSphere Application Server repository database
- ▶ WebSphere Commerce instance database
- ▶ WebSphere Commerce Payments database

Verify the runtime environment

After the databases have been restored to the remote DB2 Server, verify that the WebSphere Commerce runtime environment is working properly before proceeding with a standard store.

5.7.5 Migrate WebSphere Portal database to remote DB2 Server

After everything has been verified, the WebSphere Portal database can be migrated to the remote DB2 Server. Refer to the 5.7.4, "Migrate WebSphere Commerce databases to remote DB2 Server" on page 248 for general guidelines.

Note: This step is optional, if you are not using a remote DB2 Server for your runtime configuration.

1. Create a new instance on the DB2 Server.
2. In this case you will configure the connection from the WebSphere Portal node to the remote DB2 Server.
3. Next back up the databases and restore them on the remote DB2.
4. Catalog the databases on the WebSphere Portal node.
5. Verify the runtime environment.

5.8 Remote Web server node implementation

After a commerce enabled portal store has been verified using the WebSphere Portal node, Commerce Application Server node, and Directory node, we added a remote IBM HTTP Server for the WebSphere Portal node.

Note: For more detailed information, refer to the following:

- ▶ IBM WebSphere Portal InfoCenter, found at :
<http://publib.boulder.ibm.com/pwc/wp/42/smb/en/InfoCenter/index.html>
- ▶ *WebSphere Commerce V5.4 Handbook*, SG24-6567
- ▶ *IBM WebSphere V4.0 Advanced Edition Handbook*, SG24-6176
- ▶ *IBM WebSphere V4.0 Advanced Edition Scalability*, SG24-6192



Implement the development environment

When developing a commerce enabled portal application, it is important to consider how the various application components and development tools can be configured to provide a rich end-to-end development environment. At the core of an end-to-end development environment is the ability to develop the application code, deploy, test, and debug.

This chapter provides an explanation of the development environment that we used to develop, deploy, test, and debug the commerce enabled portal application working example. The chapter also provided detailed procedures of how to implement the development environment and debug commerce enable portal applications. We have devised and documented many unique procedures and techniques to provide source level debug and test capability.

The chapter is organized into the following sections:

- ▶ Development environment configuration options
- ▶ Prerequisite software installation
- ▶ WebSphere Commerce and WebSphere Studio Application Developer configuration
- ▶ WebSphere Portal and WebSphere Studio Application Developer configuration

6.1 Development environment configuration options

This section describes an end-to-end approach for developing and testing commerce enabled portal solutions, and includes the following topics:

- ▶ Motivation for an end-to-end development environment
- ▶ Solution overview for an end-to-end development environment
- ▶ Development environment configurations
- ▶ Hardware and software used in the development environment

6.1.1 Motivation for an end-to-end development environment

When WebSphere Commerce V5.4 was initially released in March 2002, the official development tool included with WebSphere Commerce Studio was VisualAge® for Java. In April 2003, WebSphere Commerce V5.4 officially gained support for WebSphere Studio Application Developer V5 when using WebSphere Commerce FixPak 5.4.0.5 and the corresponding WebSphere Commerce WebSphere Studio Application Developer Toolkit. When using WebSphere Studio Application Developer, developers can create, modify, and test WebSphere Commerce JSPs, commands, and EJBs within one development tool. The test environment uses the WebSphere Application Server V4.0.5 runtime within WebSphere Studio Application Developer 5.

WebSphere Portal V4.2.1 solutions are developed using the WebSphere Studio Application Developer V5. Due to some restrictions within WebSphere Studio Application Developer, a portal developer is only able to debug a portal application when the WebSphere Application Server, Advanced Single Server Edition V4 runtime is used.

The following capabilities are desirable when developing a commerce enabled portal solution:

- ▶ An end-to-end development and test scenario to develop, test, and debug commerce enabled portals including WebSphere Commerce code.
- ▶ Leverage the full WebSphere Studio Application Developer development environment capabilities to develop and test stores including HTML, commands, commerce portlets, commerce portlet JSP, and WAP WML JSPs within the WebSphere Studio Application Developer Test Environment.
- ▶ Leverage the full WebSphere Application Developer Studio to develop, test, and debug portlets.
- ▶ Simulate single sign-on to leverage the Commerce Enhancement Pack functionality.

Note: The WebSphere Studio Application Developer 5, WebSphere Application Server V4.0.5 runtime environment does not support WebSphere security settings and single sign-on services.

The interaction flow in the commerce enabled portal end-to-end development solution is summarized as follows:

1. The portal developer debugs the portal site using WebSphere Studio Application Developer by using hooks into a commerce portlet for debugging.

Note: The components of WebSphere Commerce development environment and WebSphere Portal development environment can be installed on the same machine or separate machines.

2. The portlet communicates with the WebSphere Test Environment of WebSphere Studio Application Developer while sending an HTTP URL controller command to port 8080.
3. The WebSphere Test Environment receives the portlet request. The environment is configured to stop if a request is recognized. The debugger is started and the WebSphere Commerce developer is now able to debug the code. The JSP debugger can be used by starting the tool and setting break points in advance of the request for the JSP.
4. The URL controller command redirects to a JSP, which renders the appropriate markup language. The JSP debugger can be used to follow the generation process.
5. When the JSP compilation and execution completes, the response is passed to the portlet.
6. The portal developer debugs the WebSphere Commerce response and controls the portal rendering.

6.1.2 Solution overview for an end-to-end development environment

This section provides detailed information for identifying which changes are necessary to achieve an end-to-end test and debug of commerce enabled portal solutions. A brief overview of the solution is illustrated in Figure 6-1 on page 256.

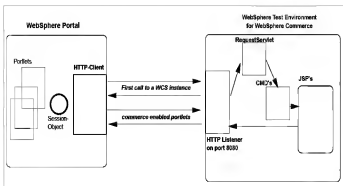


Figure 6-1 Enabling end-to-end testing

WebSphere Commerce

The developer is able to develop customized code and JSPs provided by WebSphere Studio Application Developer. After installing and configuring the Commerce Enhancement Pack, a developer is now able to develop commerce enabled JSPs as well as business logic to serve HTML/WML portal request. The WebSphere Test Environment remains unchanged.

Security and single sign-on services are not supported within the WebSphere Studio Application Developer Test Environment (WebSphere Application Server V4.0.5, Advanced Single Server Edition) but can be simulated. Therefore, the WebSphere Commerce developer has to create users within the WebSphere Commerce environment. In general, the developer uses the WebSphere Commerce Administrator Console.

User authentication is based on the WebSphere Commerce services. WebSphere Commerce offers the **Logon URL** controller command to authenticate a user by the designated `logonId` and `logonPassword`. The command creates a set of WebSphere Commerce cookies that authenticates/identifies the user.

The WebSphere Commerce **Logon URL** controller command shown in Example 6-1 is one line.

Example 6-1 Logon URL command

```
http://<hostname>:8080/webapp/wcs/stores/servlet/Logon?loginId=wcsadmin&loginPassword=password0&URL=http://host_name:8080/webapp/wcs/stores/dummy.html&storeId=10051&reLogonURL=http://host_name:8080/webapp/wcs/stores/dummy.html
```

WebSphere Portal

A portal developer uses WebSphere Studio Application Developer to develop and test portlets. In order to test commerce enabled portlets a WebSphere Portal environment must be configured properly with the Commerce Enhancement Pack while disabling the single sign-on. Since some WebSphere Commerce URL controller commands force user authentication a **Logon URL** controller command is required to authenticate the portal user.

Note: For each WebSphere Portal user, a corresponding WebSphere Commerce user must be created with the same user ID and password settings.

Whenever a WebSphere Portal user uses a commerce enabled portal application, the instance data is associated to the user session. The commerce enabled portal solution manages a `SessionObject`, which is shared by all commerce portlets of the WebSphere Portal user. This `SessionObject` provides services to maintain session relevant data to issue a WebSphere Commerce HTTP request. The commerce portlet internally sends a WebSphere Commerce HTTP requests using the `HttpClient`.

To authenticate a WebSphere Portal user with WebSphere Commerce, we have to send once and only once a **Logon URL** controller command to fetch the WebSphere Commerce cookies. Once we received the cookies, they are handled by the `SessionObject` as part of the commerce enabled portal solution.

Commerce enabled portlets share the user-based cookies to send authenticated WebSphere Commerce HTTP requests.

Summary

The approach presented enables developers with an environment to develop, test and debug WebSphere Portal and WebSphere Commerce applications in a commerce enabled portal solution. The advantages are as follows:

- ▶ Simulation of single sign-on services provided by the ITSO redbook sample, without changing the commerce enabled portal functionality included in the IBM Commerce Enhancement Pack - April 2003 Edition.
- ▶ End-to-end commerce enabled portal development and test.
- ▶ Develop, test and debug portal HTML/WAP WML JSPs and business logic using WebSphere Studio Application Developer.
- ▶ Develop, test, and debug portlets using WebSphere Application Developer Studio.

6.1.3 Development environment configurations

There are many different possibilities for configuring the development environment components. We have described four different configurations that highlight the key issues and limitations:

- ▶ Single-node development environment
- ▶ Multi-node development environment

Single-node development environment

As the name implies, all development tools and unit test runtime environment components are installed on a single node. The advantage of this type of configuration is that it is self-contained and makes deployment of the development assets to the runtime environment easier. The biggest disadvantage is the vast memory requirement (2 GB recommended, 1.5 GB minimum).

Note: The ITSO working example was created using the WebSphere Commerce Portal single-node development environment illustrated in Figure 6-2 on page 259.

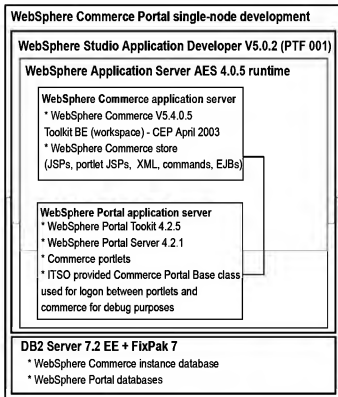


Figure 6-2 WebSphere Commerce Portal single-node development environment

Figure 6-2 depicts a single-node development environment where all development tools and supporting components are installed on the same node.

Note: This chapter includes a detailed example on how to implement the single-node WebSphere Commerce Portal development environment.

Multi-node development environment

A multi-node development environment includes a development node for WebSphere Commerce and separate development node for WebSphere Portal.

The advantage of this type of environment is reduced system memory requirements for each node. The disadvantage is that you will now need more

than one system for the development environment. Also, the deployment of application assets being developed becomes slightly more complicated.

Figure 6-3 depicts a WebSphere Commerce Portal multi-node development environment.

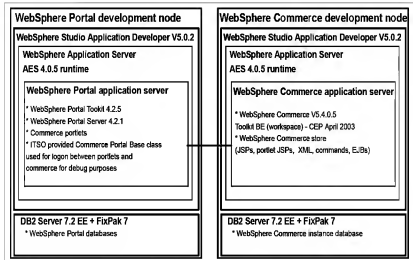


Figure 6-3 WebSphere Commerce Portal multi-node development environment

Team runtime environment for testing

After the unit testing has been completed within the previously described development environments, we recommend that you deploy and test your commerce enabled portal in a fully functional commerce enabled portal runtime environment that includes a directory server and single sign-on. This is important for several reasons. First, there are certain features developed that cannot be tested until they are deployed in the full runtime environment. Second, a team runtime environment server promotes working out integration issues with software created by other developers.

For details on implementing a commerce enabled portal runtime environment, refer to Chapter 5, "Implement the runtime environment" on page 165.

6.1.4 Hardware and software used in the development environment

This section describes the hardware and software used by the ITSO for the single node development environment.

Hardware used for the ITSO development environment

Depending on the capability and capacity of your development hardware, all components can be installed on one node. In our example, we installed the development environment software on a single node.

- IBM NetVista™ PC (6792-MHU)
 - 1 CPU, Intel P4 1.8 GHz
 - 1 GB memory
 - 40 GB hard disk
 - Intel® PRO/100+ PCI Adapter

Software used for the ITSO development environment

There are many possible configurations for a WebSphere Commerce Portal development environment. Table 6-1 lists the components needed for a WebSphere Commerce Portal development environment and the memory used by each component. The development environment includes a unit test runtime environment where the development assets can be tested as well as application development tools.

Table 6-1 Component summary

Software component	Version
Windows 2000 Server	Service Pack 3
DB2 UDB V7.2 Enterprise Edition and FixPak 7 (Server)	V7.1.0.68
WebSphere Studio Application Developer * WebSphere Studio Application Developer V5 * WebSphere Studio Application Developer PTF 001 * WebSphere Application Server AES FixPak V4.0.5	V5.0.2
WebSphere Commerce Portal Toolkit	V4.2.5
WebSphere Portal Server	V4.2.1
IBM Commerce Enhancement Pack - April 2003 Edition Toolkit for WebSphere Studio Application Developer V5	V5.4.0.5 April 2003

6.2 Prerequisite software installation

This section describes the prerequisite software to be installed prior to installing the WebSphere Commerce and WebSphere Portal Toolkits.

6.2.1 DB2 Server installation

For detailed information on installing the DB2 Server to be used by the WebSphere Commerce instance, refer to the *WebSphere Commerce V5.4 Handbook*, SG24-6567 or the *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Windows*.

The high-level installation steps are as follows:

- ▶ Install DB2 Server 7.2
- ▶ Install DB2 7.2 FixPak 7
- ▶ Configure JDBC2

6.2.2 WebSphere Studio Application Developer installation

Refer to the *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition* for details.

During the installation, we made the following selections and changes:

- ▶ Add WebSphere Application Server 4 runtime support
- ▶ Installation path: c:\ibm\wsad

WebSphere Studio Application Developer 5 PTF 001 General Fixes installation

To install the WebSphere Studio Application Developer PTF 001 - General Fixes, refer to the following:

- ▶ WebSphere Studio Application Developer 5 PTF 001 (WebSphere Studio Application Developer 5.0.0.2) is different than WebSphere Studio Application Developer 5 PTF 1 (WebSphere Studio Application Developer 5.0.1).

Important: At the time of writing this redbook, WebSphere Studio Application Developer V5 PTF 1 and WebSphere Portal Toolkit 4.2.5 do not work properly together. This problem has been reported and is being worked on by the WebSphere Portal development and support organization.

- ▶ Download WebSphere Studio Application Developer 5 PTF 001 General Fixes from the following:

http://www-1.ibm.com/support/docview.wss?rs=457&context=SS8RLP&q=&id=swg24004419&loc=en_US&cs=utf-8&lang=en

- ▶ Refer to PTF installation readme about starting location (working folder) or workspace. Change to c:\workspace.

WebSphere Studio Application Developer - WebSphere Application Server FixPak 4.0.5 install

The WebSphere Portal Toolkit V4.2.5 and WebSphere Portal Server V4.2.1 required WebSphere Studio Application Developer V5 - WebSphere Application Server AES FixPak V4.0.5. Refer to the following for details:

- Installation instructions can be found for installing the WebSphere Application Server AES FixPak V4.0.5 at:

http://www3.software.ibm.com/ibmdl/pub/software/websphere/studiotools/html/501/wsad/install_was.html

- WebSphere Studio Application Developer - WebSphere Application Server FixPak 4.0.5 needed for WebSphere Portal 4.2.1

Configure Web browser for WebSphere Studio Application Developer

You must configure the Web browser with WebSphere Studio Application Developer to use an external Web browser.

Important: If you do not configure the WebSphere Studio Application Developer to use an external Web browser, you will experience an access control problem when attempting to run Store Services from within WebSphere Studio Application Developer.

On a typical development system, there are two versions of Internet Explorer (IE):

- ▶ IE 6 installed on the system
- ▶ IE installed as part of WebSphere Studio Application Developer

When starting Store Services from the Web browser in WebSphere Studio Application Developer 5, it opens a new window using the IE 6 installed on the system. This new IE window is a new session (different browser); thus you get the access control error message.

To work around this problem, do one of the following:

1. Update WebSphere Studio Application Developer 5 to use the external browser (preferred option)

By updating WebSphere Studio Application Developer 5 to use the external IE browser, a user can click the Web browser icon from WebSphere Studio Application Developer and it will launch the IE 6 from the system. In the Store Services scenario, this will be the same browser (external). Store Services works correctly.

2. The alternative is to just use the external Internet Explorer (do not use the IE built-in Web browser of WebSphere Studio Application Developer 5).

Configuring the WebSphere Studio Application Developer for editing large XML files

While writing this redbook, we discovered that opening a large WebSphere Commerce catalog XML data file with the XML editor included with the WebSphere Studio Application Developer V5 causes WebSphere Studio Application Developer to terminate. The problem seems to be with the XML Editor design view.

This problem has been reported and the following workaround has been provided.

1. From the main menu in WebSphere Studio Application Developer, select **Windows -> Preferences**.
2. Select **Workbench -> File Associations**.
3. Select the *.xml file.

4. From the Associated editors list, click **Add** and the **Source Editor**.
5. Click **Default** to make the Source Editor the default editor.

Now when opening an XML file, the Source Editor will be used by default. The Design View and several toolbar icons will be absent, but the editor will still provide an outline and source view content assist.

6.2.3 Jakarta Ant overview

Another benefit of using WebSphere Studio Application Developer is its ability to work with third-party components such as Jakarta's Ant. Ant is a powerful build tool that assists users in the creation, packaging, and deployment of J2EE Enterprise Applications. We used Ant in our sample scenario to automate the creation and deployment of the sample store archive. We modified the necessary XML configuration files for the store archive and then configured Ant to package and deploy the end result. Ant is shipped with WebSphere Studio Application Developer and already configured and ready to use when you start up your workspace.

For information on Ant, refer to the following URL:

<http://Jakarta.apache.org/ant>

6.3 WebSphere Commerce and WebSphere Studio Application Developer configuration

This section describes how to install and configure the WebSphere Studio Application Developer V5 with the WebSphere Commerce V5.4.0.5 Toolkit provided in the IBM Commerce Enhancement Pack - April 2003 Edition distribution available for download.

6.3.1 Prerequisite software

In our example, we are using a single-node development environment configuration. Prior to installation, WebSphere Commerce requires that the following be installed:

- ▶ **DB2 Server**
For details refer to 6.2.1, "DB2 Server installation" on page 262
- ▶ **WebSphere Studio Application Developer V5**
For details refer to 6.2.2, "WebSphere Studio Application Developer installation" on page 262

6.3.2 Commerce Enhancement Pack Toolkit installation

To install the IBM Commerce Enhancement Pack Toolkit for WebSphere Studio Application Developer, do the following:

1. For more information on the IBM Commerce Enhancement Pack - April 2003 Edition, refer to the following URL:

<http://www-3.ibm.com/software/webservers/commerce/epacks/v54/>

2. You must be registered to log on in order to download the Toolkit.
3. We downloaded and unpacked the following zip file to the c:\temp\cep directory:

TLKT_BE_5405_OCT02_CommerceEnhancementPack.zip

Note: The name of the zip file may be confusing to some. The file name includes OCT02, yet it is part of the IBM Commerce Enhancement Pack - April 2003 Edition distribution.

4. You will have the following files after unpacking the TLKT_BE_5405_OCT02_CommerceEnhancementPack.zip:
 - Enhancements.pdf (*WebSphere Commerce Studio and WebSphere Studio Application Developer Integration Guide V5.4*)
 - Setup.exe (Toolkit)
5. Refer to the *WebSphere Commerce Studio and WebSphere Studio Application Developer Integration Guide V5.4* (Enhancements.pdf) for details on how to install the Toolkit.

6.3.3 Create the WebSphere Commerce development instance

After you have installed the WebSphere Commerce Toolkit for WebSphere Studio Application Developer, you will need to create a WebSphere Commerce instance within the development environment.

For details refer to the *WebSphere Commerce Studio and WebSphere Studio Application Developer Integration Guide V5.4*.

The high-level steps are as follows:

- ▶ Create/select WebSphere Studio Application Developer workspace and save preferences. This will import the workspace.
- ▶ Create the WebSphere Commerce instance from a command-line script.
- ▶ Verify instance creation (instance database logs, instance deployment logs).

6.3.4 Publishing stores using Store Services (seed environment)

After the WebSphere Commerce instance has been created and verified, you will need to publish a store archive to “seed” the environment for development.

Refer to the *WebSphere Commerce Studio and WebSphere Studio Application Developer Integration Guide V5.4* for details.

We have listed the key issues we identified when following the document process for publishing a store using Store Services with WebSphere Studio Application Developer.

- ▶ When attempting to log on to Store Services within the WebSphere Studio Application Developer using the built-in Web browser, we received the Error 0: _ERR_USER_AUTHORITY error message. To work around this problem, refer to “Configure Web browser for WebSphere Studio Application Developer” on page 263.
- ▶ After logging on to Store Services, you will have to wait until the page is compiled before it is displayed (can take a couple of minutes). Use the Web browser refresh option.
- ▶ When creating a new store within Store Services, you may see the alert message This page contains both secure and non-secure items. Do you want to display the non-secure items? We selected Yes.

6.4 WebSphere Portal and WebSphere Studio Application Developer configuration

This section describes how to install and configure WebSphere Portal V4.2.1 and supporting Toolkits within WebSphere Studio Application Developer V5.

6.4.1 Prerequisite software

In our example, we are using a single-node development environment configuration. Prior to installation, WebSphere Portal requires the following to be installed:

- ▶ DB2 Server
For details refer to 6.2.1, “DB2 Server installation” on page 262
- ▶ WebSphere Studio Application Developer V5
For details refer to 6.2.2, “WebSphere Studio Application Developer installation” on page 262

6.4.2 WebSphere Portal Toolkit V4.2.5 and WebSphere Portal V4.2.1 test environment installation

To install the WebSphere Portal Toolkit V4.2.5 (Web download) and WebSphere Portal V4.2.1 (included in WebSphere Portal V4.2.1 Enable distribution) test environment, do the following:

1. For information and download of the WebSphere Portal Toolkit V4.2.5, refer to the following URL:

<http://www-3.ibm.com/software/infocenter/websphere/index.jsp?tab=products/portal/toolkit>

2. Run **PortalToolkit425.exe** to extract the installation files.
3. Close WebSphere Studio Application Developer (if it is already started) prior to WebSphere Portal Toolkit installation.
4. Ensure DB2 is started.
5. To start the install the WebSphere Portal Toolkit and WebSphere Portal Server within the WebSphere Studio Application Developer Test Environment, run **install.bat** from the directory where it was extracted.
6. Select **English** and click **OK**. Then accept the license agreement.

Note: Language selection may not appear if your operating system locale is set to English.

7. On the component selection window, select **Portal Toolkit** (the default) and **WebSphere Portal for Test Environment**.

Note: If the machine has an existing installation of WebSphere Portal on WebSphere Application Server Single Server Edition, the WebSphere Portal for Test Environment cannot be installed by the Portal Toolkit installer. The Toolkit will use the existing WebSphere Portal installation for debugging.

Proceed to install Portal Toolkit 4.2.5.

8. If **WebSphere Portal for Test Environment** check box was selected, the WebSphere Portal installation parameters window will be displayed. Enter the required information as shown in Figure 6-4 on page 269. Make sure the fully qualified host name is entered and can be resolved from your machine. We recommend running the WebSphere Portal Test Environment on port 9080 to prevent possible conflicts with WebSphere Commerce development environment which may use port 80. Click **Next**.

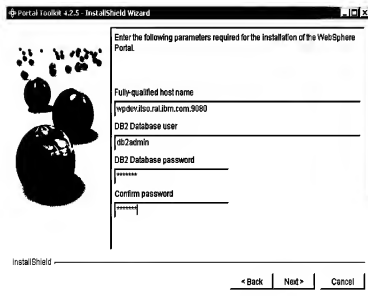


Figure 6-4 WebSphere Portal Test Environment installation settings

9. Enter the location of the wps and pzn subdirectories from WebSphere Portal 4.2.1 CD7. Click **Next** and proceed with the WebSphere Portal installation.
10. After the installation completes, click **Next**, then **Finish** to exit the installation program.
11. Start WebSphere Studio Application Developer.
12. When WebSphere Studio Application Developer starts, you will be asked if you want to open the Update Manager. Click **Yes**.
13. Pending Configuration Changes window will open as shown in Figure 6-5 on page 270. Select the top level check box and click **Finish**.

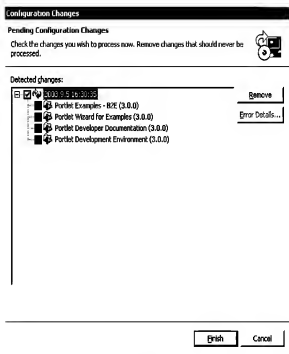


Figure 6-5 Configuration selection window

14. Click **Yes** when prompted to restart the workbench.

6.4.3 WebSphere Portal test environment server configuration

To configure the WebSphere Portal WebSphere Studio Application Developer test environment, do the following:

1. Create a WebSphere Portal server instance and configure the test environment.
 - a. When WebSphere Studio Application Developer restarts, open the Portlet perspective.
 - b. In the Server Configuration pane, right-click and select **New -> Server** and **Server Configuration** to create a new server.

- c. Select **WebSphere Portal version 4.2 -> Test Environment** server type. Enter a server name for this server instance (for example, WP 4.2.1 Test Server) and click **Next**.
- d. If you have not previously defined any servers in the Servers folder in this workspace, you will be prompted to create a new server project with the name *Servers*. Click **Yes**.
- e. Verify that HTTP port number matches the port number specified during Portal Toolkit installation in Figure 6-4 on page 269. If you used a port other than 9080, enter the port that you used. Click **Finish**.
- f. Expand the **Servers** project in Server Configuration pane and double-click **WP 4.2.1 Test Server** (newly created server) to open it in the server configuration editor.
- g. Select the **Port** tab in the server configuration editor. Remove entries for ports 80 and 443 from the host alias list and HTTP transport list. This is necessary to prevent port conflicts with WebSphere Commerce development environment if it will be installed on the same machine.

Refer to Table 6-2 for details on the ports we used for WebSphere Commerce and WebSphere Portal within the Advanced tab.

Table 6-2 WebSphere Studio Application Developer port settings

Advanced ports	WebSphere Commerce port	WebSphere Portal port
Object level trace port	2102	2112
Location server daemon port	9000	9010
Trace service port	7000	7010
Naming service port	9070	
Orb boot strap port	900	910
Admin Test port		

- h. Select the **Portal** tab in the server configuration editor. Change the WebSphere Portal debug user ID to `wsdebug` and the password to `wsdebug`.

Note: The `wsdebug` user needs to be created for both the WebSphere Commerce and WebSphere Portal within the Test Environment.

- i. Press Ctrl+S to save the server configuration. Close the server configuration editor.

You can also save by clicking **File -> Save** option or by clicking the **Save Editor Content** icon from the tool bar.

Note: In order to successfully start the WebSphere Portal test environment server, at least one portlet project needs to be created and added to the server. If no portlet projects are defined, the server will exit immediately after it starts up.

2. Create a portlet project to test WebSphere Portal installation.
 - a. In the J2EE Navigator pane, right-click and select **New -> Project**.
 - b. Select **Portlet Development -> Portlet Application Project**. Click **Next**.
 - c. Enter the project name (for example, **first test**). By default the portlet will be created in the **DefaultEAR Enterprise Application project**. Click **Next**.
 - d. Select **Basic Portlet** and click **Next**.
 - e. On next window, click **Finish**.
 - f. Stop the portal server instance if it is running. To stop the server, select the **WP 4.2.1 Test Server** from the Servers pane.
 - g. Right-click and select **Stop** or click the **Stop the server** icon from the tool bar on the Servers pane while the server instance is selected.
 - h. In the Server Configuration pane, right-click **WP 4.2.1 Test Server**. Select **Add -> DefaultEAR**.
 - i. In J2EE Navigator pane, right-click the **first test** portlet project. Select **Run on Server**.
 - j. After the server starts, you will see the following message in the Web browser window: "The system could not log into your account". At this point we have not created the debug user. We will create the same **wsdebug** user in the next step.
 - k. Click the **Sign up** button on the portal browser page to register the **wsdebug** user. It will open a register form. Fill in all the required fields in the form. In ITSQ, we used the user ID **wsdebug** and password **wsdebug**. See details in Figure 6-6 on page 273. After inserting all required information, click the **Continue** button.

IBM WebSphere Portal - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites

Address http://localhost:4040/portal/ibm.com/3000/portal/ond/PrepareEnrollment Go Links

WebSphere Portal - Microsoft Internet Explorer Register password Sign up Help Log in

Welcome, please enter your information. Continue Cancel

This information helps to personalize the content that you see. If you have any concerns, read our [Privacy Policy](#).

* User ID:
jepsdebug

* Password:
password

* Confirm Password:
password

* First Name:
WPS

* Last Name:
Debug

Email:

Preferred language:
[nothing selected]

Interests:
[nothing selected]

Sex:
[nothing selected]

Age Group:
[nothing selected]

*Required Field

Figure 6-6 Portal Debug user creation form

- l. In the Review Personal Information window, check the information and click the **Continue** button. In the User Creation Status window, click the **Continue** button. On the next page you will see the message "There is no content available".
- m. Restart the WP 4.2.1 Test Server. Now select the first test project in the J2EE Navigator pane and right-click and select **Run on Server**. WebSphere Studio Application Developer will open a portal page with the first test portlet in a Web browser. The Run on Server option always starts the server first if it is not started.

6.5 Integrating WebSphere Portal and WebSphere Commerce within WebSphere Studio Application Developer

Now that WebSphere Commerce and WebSphere Portal are installed and configured within WebSphere Studio Application Developer, we will describe how to integrate the two servers within the test environment for debug purposes.

6.5.1 Extract SSL certificate from WebSphere Commerce development environment

The following steps need to be performed on the machine where WebSphere Commerce development environment is installed:

1. In a command window, go to the <WSAD_HOME>/runtimes/aes_v4/etc directory. In our example that is c:\ibm\wsad\runtimes\aes_v4\etc.
2. Start the IBM key management utility by running:

```
..\bin\ikeman
```
3. Click **Key Database File -> Open**. Click **Browse** and select **DummyKeyring.jks**. If you have configured WebSphere Application Server to use a SSL certificate other than the default, select the keyring file where your certificate is located and click **Open**. Click **OK**. Enter the keyring password. The default password for the DummyKeyring.jks file is webAS. Click **OK**.
4. Select **WebSphere dummy server certificate**. Click **Extract Certificate...**
5. Fill in the following fields in the pop-up window:
 - Data type: Select **Base64-encoded ASCII data**
 - Certificate file name: **wc_was_cert.arm**
 - Location: **C:\temp**
 - Click **OK** to extract the certificate
6. Close the IBM Key Management Utility.

6.5.2 Enable SSL for WebSphere Portal and import the SSL certificate from WebSphere Commerce to WebSphere Portal

1. To enable SSL and import the certificate, do the following:
 - a. Extract the certificate from the WebSphere Commerce application server.

- b. Import the certificate.

Note: For details, refer to 6.5.1, "Extract SSL certificate from WebSphere Commerce development environment" on page 274.

2. Enable the IBM Secure Sockets Extension API by editing the security properties file
<WSAD_HOME>\runtimes\aes_v4\java\ire\lib\security\java.security on the WebSphere Portal node.
3. Verify that the following line exists and add it if necessary:
`security.provider.x=com.ibm.jsse.JSSEProvider`
Where *x* is the sequence number (for example, 2 or 3).
4. Import the certificate from WebSphere Commerce (application server) to the WebSphere Portal Server.
 - a. Copy the `wc_was_cert.arm` file from the `C:\temp` folder in WebSphere Commerce development system to
<WSAD_HOME>\runtimes\aes_v4\java\ire\lib\security folder in the WebSphere Portal development system.
 - b. Change to the <WSAD_HOME>\runtimes\aes_v4\java\ire\lib\security directory and enter the following command:

```
..\..\bin\keytool -import -alias MCCERT -file wc_was_cert.arm -keystore cacerts
```

This command will import the SSL certificate into the default `cacerts` keystore. The default password for the `cacerts` keystore is `changeit`. An alias identifies this certificate in the keystore and it can be any name you choose.
 - c. You will be prompted to trust the certificate (self-signed). Enter `yes`.

6.5.3 Commerce Enhancement Pack personalization installation on WebSphere Studio Application Developer 5 WebSphere Portal Test Environment (optional)

This section describes how to install the Commerce Enabled Portal personalization functionality included with the IBM Commerce Enhancement Pack - April 2003 Edition, on the WebSphere Studio Application Developer V5 with WebSphere Portal Toolkit V4.2.5 and WebSphere Portal V4.2.1 Test Environment. This procedure is optional if you do not plan on using personalization.

To install the Commerce Enhancement Pack in WebSphere Studio Application Developer V5 Test Environment, perform the following high-level steps on the WebSphere Portal development test node. In our example it is the same machine as WebSphere Commerce development system.

Note: We were not able to get all of the personalization functionality documented in the following procedure working within the WebSphere Studio Application Developer test environment. We left the instructions in place as a head start if you are interested in working your way through this (debug).

1. After the release of IBM Commerce Enhancement Pack - April 2003 Edition, a commerce enabled portal APAR zip file (WIN_JR18068.zip) was made available for download. To install the APAR files, do the following:

Enter the following URL and download the commerce enabled portal WIN_JR18068.zip file to c:\temp.

http://www-1.ibm.com/support/docview.wss?rs=494&context=SSZLC2&q=4.2&uid=swg21107559&loc=en_US&cs=utf-8&lang=en

Important: Do not use the procedure in the Readme to deploy the APAR files in the WebSphere Studio Application Developer test environment.

The `install_wc.bat` command is designed to install the APAR on a WebSphere Commerce runtime node or WebSphere Commerce Studio (VisualAge for Java). In our example, the WebSphere Commerce development environment is using WebSphere Studio Application Developer. For this reason, we must copy an "APAR serviced" directory from a WebSphere Commerce runtime node where the APAR has been installed.

In 5.2.15, "Commerce Enabled Portal - APAR JR18068" on page 198, we installed the APAR to the WebSphere Commerce runtime. We will copy this service directory to the WebSphere Portal development node.

2. On the WebSphere Commerce runtime, the IBM Commerce Enhancement Pack - April 2003 Edition was unpacked to the c:\temp\cep directory. The APAR was installed to the WebSphere Commerce runtime node and then files were copied to the c:\temp\cep (location of Enhancement Pack files) as per instructions in the APAR readme.
3. Copy the APAR updated c:\temp\cep directory (and subdirectories) on the WebSphere Commerce runtime node to the c:\temp\cep directory of the WebSphere Portal development node.

4. If the WebSphere Portal server is running, stop the WebSphere Portal server that was created in 6.4.3, "WebSphere Portal test environment server configuration" on page 270.
5. Install the Commerce Enabled Portal Enterprise Application database.
 - a. Open a command prompt and change to the `C:\temp\CEP\Base\PersonalizationUserHomePageBaseFolder\scripts` directory.
 - b. Create the CPS database by running the `createCPSDB.bat` file at the command prompt. You will be prompted with the following:
 - Enter DB2 Admin UID [db2admin]: `<db2admin_user>`
If your database user ID is same as `db2admin` then press Enter.
Otherwise, type the correct user ID for the DB2 admin and press Enter.
 - Enter Your Host Name [host]: `<wasaes_hostname>`
If your host name is `host` then press Enter. Otherwise, type your host name and press Enter.
 - Enter the password for `<host>\<db2admin_user>`
Now it will ask for DB2 Admin user's password. Type the password and press Enter. This will launch a DB2 command line program (CLP) session to create the CPS database.
 - c. When the process completes, review the `createCPSDB.log` file found in the `c:\temp\CEP\Base\PersonalizationUserHomePageBaseFolder\logs` directory.
6. Create a CPS data source in WebSphere Studio Application Developer for the WebSphere Portal Test Environment.

Note: To create a data source, the IBM Commerce Enhancement Pack - April 2003 Edition provides the `createCPSDS.db2.bat` script file, which only works if you installed WebSphere Application Server Advanced Edition and used the XML loader.

The following procedure describes how to create a JDBC driver data source for the WebSphere Studio Application Developer V5 WebSphere Portal Test Environment.

- a. In WebSphere Studio Application Developer Server Configuration pane, double-click the WebSphere Portal server (for example, WP 4.2.1 Test Server) definition to open the server configuration editor.
- b. Click the **Data Source** tab from the Server Configuration Editor window.
- c. Click the **Add** button next to the JDBC driver list.

- d. Enter the required driver definition as shown in Figure 6-7.
- Name: CPSJDBCdriver
 - Description: CPS DB2 JDBC2-compliant Driver
 - Implementation class name:
COM.ibm.db2.jdbc.DB2ConnectionPoolDataSource
 - Click **OK**.

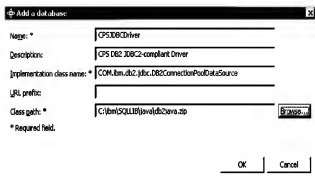


Figure 6-7 Create CPS JDBC driver

- e. Select the CPSJDBCdriver entry in the JDBC driver list. Click the **Add** button next to the data source defined in the JDBC Driver list to create a data source for this driver.
- f. Enter the data source definition as shown in Figure 6-8 on page 279. Click **OK**.

Add a Data Source

Name: *

JNDI name: *

Description:

Category:

Database name:

Default user ID:

Default user password:

Minimum pool size:

Maximum pool size:

Connection timeout:

Idle timeout:

Orphan timeout:

Statement cache size:

☒ Disable auto connection cleanup

☐ Enable JTA

* Required field.

Figure 6-8 Create a CPS data source

- g. Save the configuration by clicking the Save icon from WebSphere Studio Application Developer tool bar or pressing Ctrl+S. Close the Server Configuration Editor.
7. Install the application on the WebSphere Application Server AEs.
 - a. Create a createCPSEA_AEs.bat script file to install the CPS application in WebSphere Portal Test Environment (see Example 6-2 on page 280). Update the values of the following variables in the createCPSEA_AEs.bat:
 - NODE
 - WSAD_HOME
 - SERVER_CFG

Note: The NODE value is case sensitive. For example, if your WebSphere Application Server Administrative Server node name is all uppercase, the NODE value must be uppercase.

```
@echo off
REM WAS node name
set NODE=mka0kkcf

REM Path where NSAD V5 is installed
set NSAD_HOME=C:\ibm\wsad

REM Location of server-cfg.xml file for the Portal Server test environment
set SERVER_CFG="C:\K:\wcl\wcs.wte\WP 4.2.1 Test Server.wpc2\server-cfg.xml"

REM ===== DO NOT MODIFY CODE BELOW THIS LINE =====
set INSDIR=%NSAD_HOME%\run\times\CPS\app\cps.ear
set WASPATH=%NSAD_HOME%\run\times\aes_v4

REM CPS installation archive location
set INSPATH=..\bin\CPSEAP.ear

echo Installing Commerce Portal Server...
call %WASPATH%\bin\SEAppInstall.bat -install %INSPATH% -nodeName %NODE%
-expandDir %INSDIR% -configFile %SERVER_CFG%

set NSAD_HOME=
set SERVER_CFG=
set INSPATH=
set WASPATH=
set INSDIR=
set NODE=
```

- b. Copy the createCPSEA_AEs.bat file created in the previous step to the C:\temp\cep\Base\PersonalizationUserHomePageBaseFolder\scripts directory.
- c. Open a command prompt and change to the C:\temp\cep\Base\PersonalizationUserHomePageBaseFolder\scripts directory.
- d. Set the TEMP and TMP location to the <temp> directory where you have unpacked the WebSphereCommerceEnabledPortal.zip. For example:

```
set TEMP=C:\temp
set TMP=C:\temp
```
- e. From the command prompt, type createCPSEA_AEs.bat to run the batch file.
- f. When asked, "Do you wish to deploy all of the EJBs in this application ([Y]es/[n]o)?" type Y and press Enter (see Figure 6-9 on page 281).

```

C:\temp\cep\Bases\Personalization\UserHome\PageBase\Folder\scripts>test TMP-C:\temp
C:\temp\cep\Bases\Personalization\UserHome\PageBase\Folder\scripts>createCPSEA_AEs.bat
Installing Commerce Portal Server...
IBM WebSphere Application Server Release 4.0 AE
J2EE Application Installation Tool, Version 1.0
Copyright IBM Corp., 1997-2000

You have chosen to install this application in interactive mode.
When prompted for information, pressing ENTER without entering
any information will cause the default value (shown in [])
to be used for that property.
When prompted for information, entering F as the value will cause
the current value to be forced. This can be used to unset current
information.

Loading Server Configuration from C:\msi\msc.uto\WP 4.2.1 Test Server.wpc2\server-cfg.xml
Using Server on Node: B0T0M940
Server Configuration Loaded Successfully
Loading C:\temp\cep\Bases\Personalization\UserHome\PageBase\Folder\bin\CPSEAF.exe
Selecting Expansion Directory for EJB File
Do you wish to deploy all of the EJBs in this application <(Y/Yes/N/No)? Y

```

Figure 6-9 Running the createCPSEA_AEs.bat file and EJB deployment option

- g. Type 1 (IBM DB2) when asked which type of database you are using. Press Enter.
- h. When asked what DB Schema name you want to use for this application, press Enter (in our example, we do not have a schema).

```

C:\temp\createCPSEA>
The following Database Types are supported for EJB Deployment
-----
0) Not Applicable
1) IBM DB/2 Universal Database, Version 7.1 FP3, or higher
2) IBM DB/2 Universal Database for OS/390, Version 6
3) IBM DB/2 Universal Database for OS/390, Version 4 Release 5
4) Informix Database, Version 9.2, or higher
5) Microsoft SQL Server, Version 7, or higher
6) Oracle 8i
7) ORACLE 9i
8) SQL 92 Compliant Database Server
9) SQL 99 Compliant Database Server
10) Sybase Database, Version 11.92, or higher
11) MySQL Database, Version 3.23
12) Informix Database, Version 7.3, or higher
13) Microsoft SQL Server for Windows 2000
14) Sybase Database, Version 12.00, or higher

Which type of database are you using (optional, specify the number)?1
What DB Schema name do you want to use for this application (optional)?

Deploying All EJBs in C:\temp\cep\Bases\Personalization\UserHome\PageBase\Folder\bin\CPSEAF.aar to temp
vary directory.
[EJBDeploy] Cleanup the expanded directory C:\temp\CPSEAF.aar
[EJBDeploy] Deploy module com.ibm.commerce.portal.catalogservice.jar

```

Figure 6-10 Database type and schema

Note: If you did not use IBM DB/2 Universal Database, Version 7.2 FixPak 7, then follow the on-screen instructions for other options.

- i. Now it will start to deploy all required EJBs. When done, it will bind the following EJB JAR files:

- com.ibm.commerce.portal.personalization.eapEJB.jar
- com.ibm.commerce.portal.personalization.beans.CommercePortalPersonalizationContentgrpBean
- com.ibm.commerce.portal.personalization.beans.RoleacBean
- com.ibm.commerce.portal.cacheservice.jar
- com.ibm.commerce.portal.cacheservice.beans.CacheBean

For each item, enter the following and press Enter:

- JNDI Name: [Accept default value]
- Datasource JNDI name: jdbc/cps
- User ID: db2admin
- Password: <db2admin_password>

For example, in ITSO we entered the values shown in Figure 6-11.

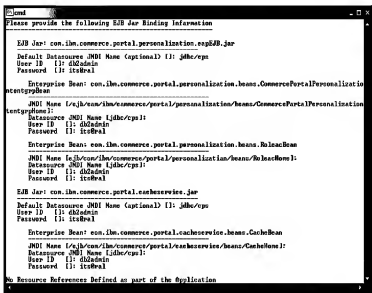


Figure 6-11 EJB bindings

- For a successful installation, you will see the information shown in Figure 6-12 on page 283.

```

C:\cmd
Password (i): its@val

No Resource References Defined as part of the Application
>>> EJB jar ...
>>> EJB jar ...
No EJB References Defined as part of the Application
No Security Roles Defined as part of Application
No "Run As Roles" Mappings defined
Installed EAR On Server
Validating Application Bindings...
Finished validating Application Bindings.
Saving EAR File to directory
Saved EAR File to directory Successfully
Backing up Server Configuration to C:\aci\acs.uts\MP 4.2.1 Test Server.wpc2\server-cfg.xml
Saving Server Configuration to C:\aci\acs.uts\MP 4.2.1 Test Server.wpc2\server-cfg.xml
Save Server Config Successful
JSP Pre-compile Skipped.....
Installation Completed Successfully
C:\temp\cep\Base\Personalization\UserHomePageBaseFolder\scripts>

```

Figure 6-12 Successful installation message

8. Run modifyWAS.bat from the command line.

When prompted to enter the WAS Path, type in the path and press Enter. It will copy three files as shown in Figure 6-13.

"WAS Path" for WebSphere Studio Application Developer test environment is <WSAD_HOME>\runtimes\aes_v4, for example
C:\ibm\wsad\runtimes\aes_v4.

9. Run installNewWPSAttributes.bat (optional) from the command line.

- When prompted to enter the path to WebSphere Application Server, enter <WSAD_HOME>\runtimes\aes_v4, where <WSAD_HOME> is the WebSphere Studio Application Developer V5 installation directory. Press Enter.
- When prompted to enter the path to WebSphere Portal Server, enter <WSAD_HOME>\runtimes\PortalServer, where <WSAD_HOME> is the WebSphere Studio Application Developer V5 installation directory. Press Enter. It will extract JSP files as displayed in Figure 6-13.

```

C:\cmd
C:\temp\cep\Base\Personalization\UserHomePageBaseFolder\scripts>modifyWAS.bat
Enter WAS Path (C:\WebSphere\AppServer\): C:\ibm\was\runtimes\aes_v4
1 file(s) copied.
1 file(s) copied.
1 file(s) copied.

C:\temp\cep\Base\Personalization\UserHomePageBaseFolder\scripts>installNewWPSAttributes.bat
Enter Path to WebSphere Application Server (C:\WebSphere\AppServer\): C:\ibm\was\runtimes\aes_v4
Enter Path to WebSphere Portal Server (C:\WebSphere\PortalServer\): C:\ibm\was\runtimes\PortalServer

extracted: NewUserConf.jsp
extracted: NewUserForm.jsp
extracted: SelfCaseUserConf.jsp
extracted: SelfCaseUserForm.jsp
extracted: UserProfileConf.jsp
extracted: UserProfileForm.jsp
..\samples\registration.properties
..\samples\registration_en.properties
2 file(s) copied.

C:\temp\cep\Base\Personalization\UserHomePageBaseFolder\scripts>

```

Figure 6-13 Scripts output

10. Verify that WebSphere Portal is functioning properly.
 - a. Start the WebSphere Portal Test Environment server instance in WebSphere Studio Application Developer.
 - b. Access the WebSphere Portal Server by entering the following URL in a Web browser:
`http://<hostname>:9080/wps/portal`
 - c. Click the **Login** button to log in and verify that you can log in as `wpsadmin`.

Note: At this stage you will not be able to log on with user IDs registered through WebSphere Commerce because there is no single sign-on capability in the development environment.

6.5.4 WebSphere Portal content publishing wizard installation (optional)

WebSphere Portal content publishing wizards can be downloaded from WebSphere Developer Domain Portal Zone at:

`http://www.software.ibm.com/wsd/zones/portal/`

Follow the instructions in Readme file to install.

6.5.5 Deploy the ITSO CEP B2B sample

After completing the configuration of the WebSphere Portal for WebSphere Commerce, we recommend that you publish the ITSO B2B CEP store to the development environment. Refer to Chapter 9, "Deploy the commerce enabled portal" on page 309 for details.

Note: For more information, refer to *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*.



Create a commerce enabled portal store

The chapter describes how to use WebSphere Studio Application Developer to import a store template, customize, package and publish the ITSO B2B CEP store.

The chapter is organized as follows:

- ▶ ITSO sample code
- ▶ Create the ITSO B2B CEP store template
- ▶ Customizing the ITSO B2B CEP store
- ▶ Package SAR and publish Web assets

7.1 ITSO sample code

This redbook provides a sample code zip file containing the artifacts developed for the redbook. These samples will be referenced throughout the remaining example chapters.

7.1.1 Procedure to download ITSO sample code

To download the sg246890.zip file, complete the following steps:

1. Enter the following URL in a Web browser:
`ftp://www.redbooks.ibm.com/redbooks/SG246890`
2. Download the sg246890.zip to a directory on your system (for example, c:\temp).
3. Unpack the sg246890.zip (for example, c:\sg246890-code).

Note: We will refer to this base unpack directory throughout the working example as c:\sg246890-code.

7.1.2 Description of ITSO sample code

The sample code zip file includes a WebSphere Studio Application Developer workspace containing all artifacts developed, ITSO-modified store archive (SAR) file, configuration files, and an ITSO-provided portlet JAR file for debug purposes.

Table 7-1 describes the contents of the ITSO sample code zip file (sg246890.zip).

Table 7-1 ITSO sample code

Directory/Filename	Description
c:\sg246890-code	ITSO sample code root directory
c:\sg246890-code\wsad	ITSO WebSphere Studio Application Developer workspace containing the ITSO B2B CEP Web assets, XML data files, and Ant scripts for packaging a SAR file and publishing.
c:\sg246890-code\sar	ITSO B2B CEP store SAR file containing ITSO-modified assets.
c:\sg246890-code\ldif	WebSphere Commerce and WebSphere Portal sample LDIF files.
c:\sg246890-code\wasaes_config	Script file used to deploy CPS application within the WebSphere Application Server AEs development environment of WebSphere Studio Application Developer.

Directory/Filename	Description
c:\sg246890-code\debug	<ul style="list-style-type: none"> ▶ ITSO-modified version of the WebSphereCommerceBasePortlet.jar to include logon support without single sign-on for debug purposes. ▶ WCSLogon.properties file containing debug WCSPassword and WCSLogonURL. <p>Note: This should only be used for debug and development, not production use. This file is not supported by IBM WebSphere Commerce.</p>

7.2 Create the ITSO B2B CEP store template

This section describes how we created the ITSO B2B CEP store template. The workspace provided in the sg246890-code.zip contains the b2bCepWeb project with these changes already completed.

7.2.1 Create a new project

Create a new project in WebSphere Studio Application Developer to import the ITSO B2B CEP store assets. This will allow for modifications to the store assets and repacking into an updated SAR file for publishing.

1. Start WebSphere Studio Application Developer and select the WebSphere Commerce workspace.
2. Select the **J2EE Navigator** tab.
3. Open the **Web perspective**.
4. Select the **J2EE Navigator** tab, right-click in the white space and select **New -> Project -> Web -> Web Project**.
5. When the Define the Web Project window appears, enter the following and click **Next**:
 - Project Name: b2bCepWeb
 - Select **J2EE Web Project**
6. On the J2EE Settings page, enter the following and click **Finish**:
 - Enterprise Application Project: select **New**
 - New Project Name: b2bCepEAR
7. Import the ITSO B2B CEP store by doing the following:
 - a. Select the newly created project (for example, b2bCepWeb). Right-click and then select **Import -> File System**.

- b. Enter the root directory of the files (in our example, c:\sg246890-code\wsad\cep\b2bCepWeb) and then click **OK**.
- c. Click **Select All** and then click **Finish**.
- d. When prompted with the message *Classpath already exists, Would you like to overwrite it?*, we clicked **No**.
- e. When prompted with the message *Web settings already exists, Would you like to overwrite it?*, we clicked **No**.

7.2.2 Update core data

The core data that was modified to create the ITSO B2B CEP store includes the following:

- Modify store.xml
- Modify the Distinguished Name (DN)

Note: When creating a template, we recommend that you update the core data files that contain information that will be common to all stores built from the template.

Refer to the *Store Developer's Guide, IBM WebSphere Commerce V5.4* or the *WebSphere Commerce V5.4 online documentation* for details on all core data assets. In addition, refer to the *WebSphere Commerce V5.4 Catalog Design and Content Management*, SG24-6585.

Modify store.xml

The core data that was updated for the ITSO B2B CEP store is located in the store.xml file. We modified the storeent_id and the store directory.

What uniquely identifies a store is a combination of the store identifier and the store owner. We modified the store identifier attribute to ensure that our store is unique (see Example 7-1 for the storeent entry). For a complete listing, please refer to the store.xml that is available in the downloaded ITSO sample workspace, found in the b2bCepWeb\sar\data directory.

Example 7-1 XML element <storeent> in store.xml (identifier=b2bCep)

```
<storeent storeent_id="@storeent_id_1"
  member_id="@MEMBER_ID;"
  type="S" identifier="b2bCep"
  setccurr="USD"/>
```

Update the directory of the store (for example, directory=b2bCep as seen in Example 7-2 on page 289).

Example 7-2 XML element <store> in store.xml (directory=b2bCep)

```
<store store_id="@storeent_id_1" directory="b2bCep"
ffmcenter_id="@ffmcenter_id_1" language_id="@en_US;" storegrp_id="-1"
allocationgoodfor="43200" bopmpadfactor="0" defaultboffset="2592000"
ffmcselectionflags="0" maxboffset="7776000" rejectedordexpiry="259200"
rtntffmctr_id="@ffmcenter_id_1" pricerefflags="0" storetype="B2B"/>
```

Note: You must edit all occurrences of the *identifier* and *directory* in the SAR directory and language dependant sub directories. For example, the en_US and es_ES must be updated, too.

Modify the Distinguished Name (DN)

There are several data files included in the ITSO B2B CEP store SAR (b2bCep_en_US_es_ES.sar) or the WebSphereB2CCommerceEnabledPortal.sar included with the IBM Commerce Enhancement Pack - April 2003 Edition, that require modifications for the Distinguished Name, including contract.xml, businessaccount.xml, and organization.xml.

The ITSO-provided WebSphere Studio Application Developer workspace (c:\sg246890\wsad\cep) contains a contract.xml.LDAP, businessaccount.xml.LDAP, and organization.xml.LDAP file with the Distinguished Name as needed for LDAP.

Note: For more detailed information, reference Technote #1083258 at:

http://www-1.ibm.com/support/docview.wss?rs=494&context=SS2LC2&q=publish&uid=swg21083258&loc=en_US&cs=utf-8&lang=en

To address this problem, do the following:

1. Check the Distinguished Name (DN) for the wcsadmin user in the USERS database table as follows:
 - a. Start a DB2 command window.
 - b. Connect to the WebSphere Commerce instance database.
db2 connect to wcsdb
 - c. Query the USERS table.
db2 select * from USERS
 - d. Record the DN for wcsadmin (for example, uid=wcsadmin,dc=ibm,dc=com).

2. Ensure that the contract.xml file in the SAR has the same DN. If the DN is different, change the DN in the contract.xml to reflect that of the wcsadmin in the database.

In our case, are using LDAP and we have a different DN for wcsadmin than the default contract.xml.

For example, you may have uid-wcsadmin,dc=ibm,dc=com. However, the contract.xml has <User distinguishName ="uid-wcsadmin,o=Root Organization" />. You need to change the contract.xml to <User distinguishName ="uid-wcsadmin,dc=ibm,dc=com" /> in order to successfully publish the Sample Commerce Portal. The contract.xml file is included within the SAR file.

Note: The b2bCepWeb\sar\data\contract.xml.LDAP in the ITSO provided workspace contains the LDAP Distinguished Name. Copy contract.xml.LDAP to contract.xml as a starting point and modify the value of the Distinguished Name as described.

7.2.3 Update configuration data

Note: For the ITSO B2B CEP store SAR, the contract.xml can be easily modified in the WebSphere Studio Application Developer project, and then repackaged using the Ant script provided called ant_create_sar.xml.

In our example scenario, we are focused on the data assets and do not need to modify the command registry, view registry, or URL registry for our sample store.

When developing a customized store, it is very common that you will create your own JSPs and commands. When this is done, you will need to update the following configuration data:

- ▶ Command registry (CMDREG table)
- ▶ View registry (VIEWREG table)
- ▶ URL registry (URLREG table)

For more information on how to update the configuration data, refer to the following:

- ▶ *WebSphere Commerce V5.4 online documentation*
- ▶ *Programmer's Guide, IBM WebSphere Commerce V5.4 product guide*
- ▶ *WebSphere Commerce V5.4 Developer's Handbook*, SG24-6190

7.3 Customizing the ITSO B2B CEP store

This section outlines the steps to enable existing JSPs for commerce enabled portlets in the ITSO B2B CEP store.

Constraints

The JSPs of a commerce enabled portal must be modified to work properly with the WebSphere Portal. There are several restrictions and constraints you have to address within the JSPs:

- ▶ JavaScript naming convention
- ▶ Form naming conventions
- ▶ Stripping HTML tags
- ▶ Enable Portal URI
- ▶ Developing commerce enabled portlets

7.3.1 JavaScript naming convention

There is a one-to-many relationship between the WebSphere Commerce Portal portlets JSPs and the commerce enabled portlets. For example, several Product Display portlets can be aggregated, each accessing the `ProductDisplay.jsp`. Within the JSP, a set of JavaScript elements can be defined. This leads to the circumstance that the JavaScript code is included multiple times within the generated HTML output of the portal page, which can create problems for a Web browser client.

If you are using JavaScript elements in the JSP, ensure that during the generation process the elements are uniquely defined. This can be solved as follows:

- ▶ Timestamp approach

Concatenate the JavaScript with a timestamp object to uniquely identify the JavaScript for the generated portlet.

Note: You can call the `Now` method from the `com.ibm.commerce.util.TimestampHelper` class to retrieve a timestamp value.

- ▶ Additional key/value pairs

Pass additional key/value pairs to the JSP (for example, portlet ID). Concatenate the JavaScript name with the value of the key object to uniquely identify the JavaScript for the generated portlet.

Timestamp approach

The modifications to enable JavaScript elements across commerce enabled portlets are done as follows:

1. Open the portal_jsp/include/GetResource.jsp file in a text editor.
2. Add the following statement:

```
int lTimeStamp = com.ibm.commerce.utils.TimestampHelper.now().getNanos();
```

Note: In order to use the lTimeStamp variable, each portal JSP must include the include/GetResource.jsp file.

3. Save the file.
4. For each JSP, do the following:
 - a. Open your JSP that contains the JavaScript function.
 - b. Rename each JavaScript function as seen in Example 7-3.

Example 7-3 Modified submitSearch JavaScript function

```
function submitSearch<%=lTimeStamp%>(form)
{
    <%= /* Show products & items if a SKU number is entered.
        * (A SKU number is unique in a store)
        * Only list items on the result page if 'manufacturer name' or
        * 'manufacturer part number' fields are input.
        */ %>
    if (form.sku.value != "")
    {
        form.resultType.value = "3"
    }
    else if (form.manufacturer.value != "" ||
        form.manufacturerPartNum.value != "")
    {
        form.resultType.value = "1"
    }
    form.submit()
}
```

- c. Save the JavaServer Page.

7.3.2 Form naming conventions

There is a one-to-many relationship between the WebSphere Commerce Portal portlet JSPs and commerce enabled portlets. For example, several Search portlets can be aggregated, each accessing the AdvancedSearch.jsp. Within the JSP a set of form elements can be defined. This leads to the circumstance that

the generated code includes multiple forms of the same name within the generated HTML output of the portal page, which can create problems for a Web browser client.

If you are using form elements in the JSP, make sure that during the generation process the form elements are uniquely defined. This can be solved as follows:

- **Timestamp approach**

Concatenate the form name with a timestamp object to uniquely identify the JavaScript for the generated portlet.

Note: You can call the `Now` method from the `com.ibm.commerce.utils.TimestampHelper` class to retrieve a timestamp value.

- **Additional key/value pairs**

Pass additional key/value pairs to the JSP (for example, portlet ID). Concatenate the form name with the value of the key object to uniquely identify the JavaScript for the generated portlet.

Timestamp approach

The modifications to enable HTML FORM tags across commerce enabled portlets are done as follows:

1. Open the `portal_jsp/include/GetResource.jsp` file in a text editor.
2. Add the following statement:

```
int lTimeStamp = com.ibm.commerce.utils.TimestampHelper.now().getNanos();
```

Note: In order to use the `lTimeStamp` variable each portal JavaServer Page must include the `include/GetResource.jsp` file.

3. Save the file.
4. For each JavaServer Page, do the following:
 - a. Open your JavaServer Page that contains the JavaScript function.
 - b. Rename each form as follows:

Example 7-4 Modified AdvancedSearchForm HTML <form> tag

```
<form method="post" action="<%=sPortalURI%>"  
name="AdvancedSearchForm<%=lTimeStamp%>">
```

- c. Save the JavaServer Page.

7.3.3 Stripping HTML tags

Since the WebSphere Portal generates the HTML/WML output for the browser, you have to strip the following HTML tags:

- ▶ <HEAD> and </HEAD> tags
- ▶ <DOCTYPE> tag
- ▶ <BODY> and </BODY> tags
- ▶ <META> tags

For more information, refer to Appendix C, "Converting Existing JSPs" in the *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*. Ironically, the B2B CEP sample provided with the IBM Commerce Enhancement Pack - April 2003 Edition does not follow the guidelines.

7.3.4 Enable Portal URI

To enable the portal URI within the commerce portlet JSP, do the following:

1. To support commerce enabled portal functionality, each portlet JSP must have the following preamble before any output is produced by the JSP file.

Example 7-5 Sample for enabling portal URI within commerce portlet JSP

```
<%
String sURLPrefix=request.getScheme()+"://"+request.getServerName()+" ";
request.getServerPort() + request.getContextPath()+"/servlet/";
String sURLimgPrefix=request.getScheme()+"://"+request.getServerName()+" ";
+
request.getServerPort();

String
sPortalURIParam=com.ibm.commerce.portal.exports.CommercePortalConstants.S_P
ORTLET_URI_PARAM;
String
sRemoteServletURIParam=com.ibm.commerce.portal.exports.CommercePortalConsta
nts.S_REMOTE_SERVLET_URI_PARAM;

String sPortalURI =null;
JSPHelper jsphelperForPrefixCode =new JSPHelper(request );
Object oPortalURI =jsphelperForPrefixCode.getParameter(sPortalURIParam );

if (oPortalURI !=null )
{
    sPortalURI =(String )oPortalURI;
    sPortalURI =sPortalURI.substring(0,sPortalURI.lastIndexOf("#"));
}
```



```

boolean fModifyLinks =false;
if (sPortalURI !=null )
    fModifyLinks =true;
}

```

Note: This code can be maintained in a JSP include file to be included by each portal_jsp JSP.

The parameters obtained in the above piece of code are used for proper link encoding by the portlet JSP and mentioned in the remaining guidelines.

2. All links in the portlet JSP file must adhere to the specific encoding convention. If the links do not follow the specified mechanism, these links will not work properly.
 - a. Image links:

If images in the JSP files are specified by absolute URLs, no changes are needed. If images in the JSP file are referenced by relative URLs, the encoding must be in the following form:

```

```

- b. URLs:

All the URLs in the JSP file must follow specific encoding schema defined for commerce enabled portlets. The following is the example of such a modification. If you had the following URL in your regular JSP file:

```

<a href="PrivacyView?langId=<%=languageId%>&
storeId=<%=storeId%>&catalogId=<%=catalogId%>">PrivacyPolicy</a>

```

The same link will be encoded in your commerce enabled Portlet JSP file as follows:

```

<a href="<%=sPortalURI%>?<%=sRemoteServletURIParam%>=
<%=java.net.URLEncoder.encode(sURLPrefix+"PrivacyView?langId="+
languageId +"&storeId="+storeId +"&catalogId="+
catalogId)%>">PrivacyPolicy</a>

```

As you can see, the difference is in the fact that now the request has to go to the <%=sPortalURI%> URL and your old URL becomes a value for the <%=sRemoteServletURIParam%> parameter.

3. Forms modification: All the forms in the JSP file must have the same value for the action, given by <%=sPortalURI%>. Also, all forms must have an additional <%=sRemoteServletURIParam%> parameter, containing the older action value of the form. If the user must see default representation of the portlet after the form submission, the following parameter must be added to the form:

```

<input
type="hidden" name="<%=con.ibm.commerce.portal.exports.CommercePortalCons

```

```
tants.S_TERMINAL_URI_PARAM}>*value="<%=com.ibm.commerce.portal.exports.C  
ommercePortalConstants.S_TERMINAL_URI_PARAM_TRUE%>">
```

For example, if your original form looked like:

```
<FORM METHOD="POST" NAME="Logon" action="Logon"><INPUT TYPE="hidden"  
NAME="storeId" VALUE="<%=storeId%>"><INPUT TYPE="hidden"  
NAME="catalogId"VALUE="<%=catalogId%>"></FORM>
```

Then it should become:

```
<FORM METHOD="POST" NAME="Logon" action="<%=sPortalURI%>"><INPUT  
TYPE="hidden" name="<%=sRemoteServletURIParam %>" value="<%=sURLPrefix  
+"Logon"%>"><INPUT TYPE="hidden" NAME="storeId"  
VALUE="<%=storeId%>"><INPUT TYPE="hidden" NAME="catalogId"  
VALUE="<%=catalogId%>"></FORM>
```

7.3.5 Developing commerce enabled portlets

To develop portlets, you can use the WebSphere Studio Application Developer wizards, subclass the provided classes of the WebSphere Commerce Portal provided with the Commerce Enhancement Pack, or write native WebSphere Portal portlets. In the event you develop native portlets, you must ensure that the UserAgent parameter is set properly.

The Commerce Enhancement Pack portal adapters assume that the following string is added to the UserAgent information:

```
(NpsHTTPClient)
```

For example:

```
httpurlconnection.setRequestProperty("User-Agent", sAgentId +  
"(NpsHTTPClient)");
```

7.4 Package SAR and publish Web assets

This section describes how to package and publish the CEP stores. In our CEP scenario, we have several CEP stores for different B2B customers.

This section includes the following tasks:

- ▶ Create a SAR using ant_create_sar.xml
- ▶ Publish Web assets to WebSphere Commerce using ant_publish_WCS.xml

To fully take advantage of the scripts provided for publishing, it is required that the WebSphere Commerce Portal development environment is installed and configured as described in Chapter 6, "Implement the development environment" on page 253.

7.4.1 Create a SAR using ant_create_sar.xml

This section describes how the `ant_create_sar.xml` script, found in the `b2bCepWeb\ant` directory of the workspace, can be used to repackage a SAR file.

Note: You must modify the `cep.config.properties` file for your environment. It contains a set of directory settings.

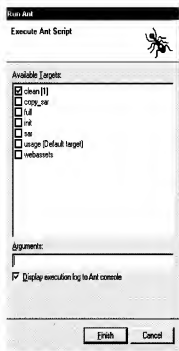


Figure 7-1 `ant_create_sar.xml` targets

To run the `ant_create_sar.xml` script within the WebSphere Studio Application Developer, do the following:

1. Open the `ant` directory under your `b2bCepWeb` folder.
2. Right-click the `ant_create_sar.xml` and from the context menu select **Run Ant**.
3. Select one of the following targets to run, as seen in Figure 7-1.

- | | |
|------------------|---|
| full | To generate all assets including the SAR file and copying the SAR file to the source directory. |
| copy_sar | To copy the SAR file to the source directory. |
| sar | To generate the SAR file including all necessary assets. |
| init | To initialize script. |
| clean | To delete all generated assets. |
| webassets | To generate all assets for the SAR file. |
4. By default, the SAR file is packaged in the
c:\sg246890-code\wsad\cep\b2bCep\Web\sar directory.

The SAR file packaged using this script can be deployed to the runtime or development test environment using Store Services as described in Chapter 9, "Deploy the commerce enabled portal" on page 309.

7.4.2 Publish Web assets to WebSphere Commerce using ant_publish_WCS.xml

The ant_publish_WCS.xml script can be used if you want to publish Web assets to the WebSphere Commerce runtime environment. This requires that you have a network mapped drive from the development node to the runtime node. The publish process updates the following directories:

- ▶ Store properties
C:\IBM\was\installedApps\WC_Enterprise_App_demo.ear\wcstores.war\WEB-INF\classes\b2bCep
- ▶ Portal assets
C:\IBM\was\installedApps\WC_Enterprise_App_demo.ear\wcstores.war\b2bCep
- ▶ Web assets
C:\IBM\was\installedApps\WC_Enterprise_App_demo.ear\wcstores.war\b2bCep
- ▶ Tools runtime xml
C:\IBM\was\installedApps\WC_Enterprise_App_demo.ear\wcstores.war\xml\b2bCep
- ▶ Tools properties
C:\IBM\was\installedApps\WC_Enterprise_App_demo.ear\properties\tools\stores\b2bCep
- ▶ Tools assets
C:\IBM\was\installedApps\WC_Enterprise_App_demo.ear\wc\tools.war\xml\tools\stores\b2bCep

Note: The instance name and directories are defined in the cep.config.properties file and cep.wcs.properties file.

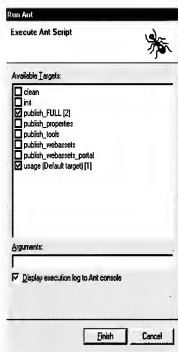


Figure 7-2 ant_publish_WCS.xml targets

To run the script within the WebSphere Studio Application Developer, do the following:

1. Open the ant directory under your b2bCepWeb folder.
2. Right-click **ant_publish_WCS.xml** and from the context menu, select **Run Ant**.
3. Select one of the following targets to run:

publish_FULL

To publish all assets to the destination directories for WebSphere Commerce.

publish_properties	To publish only the properties to the destination directories for WebSphere Commerce.
publish_tools	To publish all tool assets to the destination directories for WebSphere Commerce.
publish_webassets	To publish all Web assets to the destination directories for WebSphere Commerce.
publish_webassets_portal	To publish all portal Web assets to the destination directories for WebSphere Commerce.
init	To initialize ant_publish_WCS.xml script variables
clean	To clean publish of store assets.



Create portlets with Dynamic Context Groups

This chapter describes how to create Dynamic Context Groups for WebSphere Commerce Portal.

Portlets are by nature independent of one another. WebSphere Portal provides the portlet messaging capability such that portlets within the same portlet application can communicate with one another. WebSphere Commerce Portal extends the portlet messaging capability to achieve dynamic business content synchronization. The design of dynamic context portlet grouping provides a facility for a user to group portlets whose content is meant to be synchronized dynamically at runtime. Such a facility does not require code changes for the participating portlets within a Dynamic Context Group, but instead, Dynamic Context Grouping is done through the deployment descriptor. This greatly enhances the ease of regrouping of portlets without the need to restart the portal server as part of the regrouping exercise.

8.1 Dynamic Context Groups

A Dynamic Context Group consists of one master portlet and one or more slave portlets. Each context group contains a token (the dynamic context), which is shared among all portlets in the group. The code provided with IBM Commerce Enhancement Pack - April 2003 Edition allows only string tokens.

Note: Sharing of other data types or Java objects is not supported by IBM Commerce Enhancement Pack - April 2003 Edition.

All slave portlets must define an action that will be performed when they receive notification from the master portlet. When the master portlet detects a change in the defined dynamic context, it will signal all slave portlets to perform the predefined action. Slave portlets will pass the shared token value as an HTTP request parameter to WebSphere Commerce commands that they call. The view JSP in WebSphere Commerce that renders output for WebSphere Portal must implement logic to display the desired result based on the value of this parameter. The JSP must also provide the dynamic context value back to the portlets. More information about commerce JSP enablement for Dynamic Context Groups is found in 8.1.2, "Modifications to commerce portal JSPs" on page 305.

One portlet can belong to several context groups. It can be master in several groups, slave in several groups, or simultaneously master in some groups and slave in others.

For an example of configuring a portlet to be both a slave and a master, see `AccountRelatedPortletGroup` and `ContractRelatedPortletGroup` in the B2B Direct sample store found in the IBM Commerce Enhancement Pack - April 2003 Edition.

For more details about Dynamic Context Groups, refer to the *Commerce Enabled Portal Integration Guide*, IBM Commerce Enhancement Pack April 2003 Edition, chapter on Dynamic Context Groups.

8.1.1 Defining Dynamic Context Groups

Dynamic Context Group attributes are defined in the `portlet.xml` portlet application descriptor file as configuration parameters within `<concrete-portlet>` stanzas.

Master portlet

The master portlet for a context group is defined by the `master_list` parameter. The value of the `master_list` parameter is the context group name. If a portlet is a master in several groups, the parameter value is a comma-separated list of group names.

Example 8-1 defines the Drills portlet as a master for the `DrillsGroup_productId` context group. The Dynamic Context Group specific part is shown in *italics*.

Example 8-1 Master portlet definition for DrillsGroup_productId group

```
<concrete-portlet href="#com.ibm.commerce.portal.B2BDemo.Portlet4">
  <portlet-name>Drills</portlet-name>
  <default-locale>en</default-locale>
  <language locale="en">
    <title>Drills</title>
    <title-short>Drills</title-short>
    <description>The Drills Portlet</description>
    <keywords>WCS</keywords>
  </language>
  <config-param>
    <param-name>ApplicationNamespace</param-name>
    <param-value>com.ibm.commerce.portal.B2BDemo</param-value>
  </config-param>
  <config-param>
    <param-name>SSOEnabled</param-name>
    <param-value>true</param-value>
  </config-param>
  <config-param>
    <param-name>RefreshPriority</param-name>
    <param-value>5</param-value>
  </config-param>
  <config-param>
    <param-name>InitialRemoteURL</param-name>

    <param-value>http://wcserv1.itso.ral.ibm.com/webapp/wcs/stores/servlet/Category
    Display?catalogId=10051&storeId=10051&categoryId=10058&langId=-1&an
    p;parent_category_rn=10052&portal_subcat=true</param-value>
  </config-param>

  <config-param>
    <param-name>master_list</param-name>
    <param-value>DrillsGroup_productId</param-value>
  </config-param>
</concrete-portlet>
```

Slave portlets

Slave portlets for a context group are defined by the `slave_list`, `<context_group_name>_slave_action_url` and `<context_group_name>_context_to_param_map` parameters.

- ▶ The `slave_list` parameter value defines the context group name to which this portlet belongs as a slave. If portlet is a slave in several groups, parameter value is a comma-separated list of group names.
- ▶ The `<context_group_name>_slave_action_url` defines the URL. There should be one `slave_action_url` defined for each context group that the portlet belongs to.
- ▶ The `<context_group_name>_context_to_param_map` defines the parameter name for passing the shared token. For each context group that the portlet belongs to, there should be a `context_to_param_map` url defined.

Example 8-2 defines the Drills SKUs portlet as a slave in the `DrillsGroup_productId` context group. The shared token value will be passed to WebSphere Commerce as the `productId` parameter. The Dynamic Context Group specific part is shown in *italics*.

Example 8-2 Slave portlet definition for DrillsGroup_productId group

```
<concrete-portlet href="#com.ibm.commerce.portal.B2BDemo.Portlet12">
  <portlet-name>Drills SKUs</portlet-name>
  <default-locale>en</default-locale>
  <language locale="en">
    <title>Drills SKUs</title>
    <title-short>Drills SKUs</title-short>
    <description>The Drills SKUs Portlet</description>
    <keywords>WCS</keywords>
  </language>
  <config-param>
    <param-name>ApplicationNamespace</param-name>
    <param-value>com.ibm.commerce.portal.B2BDemo</param-value>
  </config-param>
  <config-param>
    <param-name>SSOEnabled</param-name>
    <param-value>true</param-value>
  </config-param>
  <config-param>
    <param-name>RefreshPriority</param-name>
    <param-value>5</param-value>
  </config-param>
  <config-param>
    <param-name>InitialRemoteURL</param-name>
```

```

<param-value>http://wcserv1.itso.ral.ibm.com/webapp/wcs/stores/servlet/ProductD
isplay?catalogId=10051&storeId=10051&productId=10909&langId=-1&parent_category_rn=10058&catalogOnly=true</param-value>
</config-param>

<config-param>
  <param-name>slave_list</param-name>
  <param-value>DrillsGroup_productId</param-value>
</config-param>
<config-param>
  <param-name>DrillsGroup_productId_slave_action_url</param-name>

<param-value>http://wcserv1.itso.ral.ibm.com/webapp/wcs/stores/servlet/ProductD
isplay?catalogId=10051&storeId=10051&productId=10909&langId=-1&parent_category_rn=10058&catalogOnly=true&=</param-value>
</config-param>
<config-param>
  <param-name>DrillsGroup_productId_context_to_param_map</param-name>
  <param-value>productId</param-value>
</config-param>

</concrete-portlet>

```

8.1.2 Modifications to commerce portal JSPs

The commerce portal JSPs in WebSphere Commerce instance need to be modified to produce the desired output based on the parameters passed from WebSphere Portal.

Example 8-3 shows a fragment of the modified CatalogSubCategoriesDisplay.jsp file that has been enabled for the DrillsGroup_productId context group. Note that the InitialRemoteURL setting for the Drills portlet descriptor in Example 8-2 on page 304 has the portal_subcat=true parameter appended to the URL. When this parameter is present, CatalogSubCategoriesDisplay.jsp will render output for the master portlet in the context group. Otherwise it will render output for a stand-alone commerce portlet. For simplicity, the JSP code in Example 8-3 supports only the DrillsGroup_productId context group.

See the CatalogSubCategoriesDisplay.jsp file in the B2B Direct sample store supplied with IBM Commerce Enhancement Pack - April 2003 Edition for an example of supporting multiple context groups.

Example 8-3 Commerce portal JSP modification

```
<TD valign="top" align="left">
```

```

<%
    if (portal_subcat == null)
    (
        <a
href="<%=sPortalURI%>?<%=sRemoteServletURIParam%>=<%=java.net.URLEncoder.encode
( sURLPrefix + "ProductDisplay?catalogId=" + catalogId + "&storeId=" + storeId
+ "&productId=" + product.getProductId() + "&langId=" + languageId +
"&parent_category_rn=" + categoryId + "&ter=true" + "&catalogOnly=true")%>"
<%=product.getDescription().getFullImage() %>"
alt="<%=product.getDescription().getShortDescription()%>" hspace="5"
height="50" border="0" align="left">
        </a>
        <%
    )
    } else {
        <a
href="<%=sPortalURI%>?<%=sRemoteServletURIParam%>=<%=java.net.URLEncoder.encode
( sURLPrefix + "CategoryDisplay?catalogId=" + catalogId + "&storeId=" + storeId
+ "&categoryId=" + categoryId + "&langId=" + languageId +
"&parent_category_rn=" + parentCategoryId + "&ter=true&portal_subcat=true" +
"&catalogOnly=true")%>&DrillsGroup_productId=<%=product.getProductId()%>"
<%=product.getDescription().getFullImage() %>"
alt="<%=product.getDescription().getShortDescription()%>" hspace="5"
height="50" border="0" align="left">
        </a>
        <%
    }
}
</TD>

```

8.1.3 WebSphere Studio Application Developer tooling for creating Dynamic Context Groups

IBM Commerce Enhancement Pack - April 2003 Edition provides a WebSphere Studio Application Developer plug-in to assist with Dynamic Context Group definition.

Install Dynamic Context Portlet Group Assembly Tool plug-in

To install the Dynamic Context Portlet Group Assembly Tool plug-in, do the following:

1. Copy the com.ibm.commerce.portal.tooling.dynamiccontext.zip file from C:\temp\CEP\Base\DynamicContextPortletGroupAssemblyTool directory on

the WebSphere Portal node to your WebSphere Studio Application Developer workstation.

2. Unzip the file to the `WSAD_installdir\plugins` directory. Ensure that the archive directory structure is created when unzipping.
3. Edit the `WSAD_installdir\plugins\com.ibm.commerce.portal.tooling.dynamiccontext\plugin.properties` file. Update the `Schema_Path` setting to point to the location of your `Schema.xsd` file. The default location for the file is `WSAD_installdir\plugins\com.ibm.commerce.portal.tooling.dynamiccontext\Schema.xsd`. Usually there is no need to change the default location.

Note: The backslash (\) is a special escape character in Java and many other environments. If you use backslash characters in the property file in the directory path name, which is common on Windows platforms, they must be escaped with another backslash. For example:

```
Schema_Path=C:\\Program Files\\IBM\\Application  
Developer\\plugins\\com.ibm.commerce.portal.tooling.dynamiccontext\\Sche  
ma.xsd.
```

Using forward slash (/) is the preferred, more portable way.

4. Restart WebSphere Studio Application Developer.

Defining a Dynamic Context Group

To define a Dynamic Context Group, do the following:

1. Create an XML definition file for the Dynamic Context Group. Example 8-4 shows the definition of the `DrillsGroup` context group. Note that the context group name generated by the assembly tool will be in the form `<DynamicContextGroupName>_<DynamicContext>`. In this case, it will be `DrillsGroup_productId`.

Example 8-4 Dynamic Context Group XML definition

```
<?xml version='1.0' encoding='UTF-8'?>  
<DynamicContextGroups>  
  
<DynamicContextGroup>  
  <DynamicContextGroupName>DrillsGroup</DynamicContextGroupName>  
  <DynamicContextMasterPortlet>Drills</DynamicContextMasterPortlet>  
  <DynamicContext>productId</DynamicContext>  
  <DynamicContextSlavePortlet>  
    <DynamicContextSlavePortletName>Drills  
    SKUs</DynamicContextSlavePortletName>
```

```

<SlavePortletAction>http://wcserv1.itso.ral.ibm.com/webapp/wcs/stores/servlet/P
roductDisplay?catalogId=10051&storeId=1913&productId=21201101259&lan
gId=-1&parent_category_rn=21201091058&catalogOnly=true</SlavePortletAc
tion>
    <SlaveParamMapToContext>productId</SlaveParamMapToContext>
  </DynamicContextSlavePortlet>
</DynamicContextGroup>

</DynamicContextGroups>

```

-
2. In WebSphere Studio Application Developer, click **Perspective -> Show View -> Other...** Select **Commerce Portal Tooling Views -> Dynamic Context WAR file update**. Click **OK**.
 3. In the plug-in window, specify the portlet.xml file location for your portlet application in the Portlet XML field.
 4. Specify the name of Dynamic Context Group XML definition file that you created in the XML file name field.
 5. Click **OK**.
 6. The portlet.xml file will be updated with the context groups you have defined.

For more information about using the Dynamic Context Assembly Tool, refer to the *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*.



Deploy the commerce enabled portal

The chapter includes procedures for deploying the ITSO B2B CEP store working example to the runtime environment and the development test environment.

The chapter is organized as follows:

- ▶ Deploy the ITSO B2B CEP store to the runtime environment
This section describes how to deploy the ITSO B2B CEP store to the WebSphere Commerce Portal runtime environment. This procedure is useful for verifying the runtime environment and for experimenting with the sample.
- ▶ Deploy the ITSO B2B CEP store to the development test environment
This section describes how to deploy the ITSO B2B CEP store to development test environment for development and debug purposes.

9.1 Deploy the ITSO B2B CEP store to the runtime environment

After completing the configuration of the runtime environment as described in Chapter 5, "Implement the runtime environment" on page 165, the ITSO B2B CEP store can be deployed.

This section includes the following:

1. ITSO sample code
2. Database backup
3. Server startup
4. Create a new organization for the ITSO B2B CEP store
5. Create a new user and add to CEP organization
6. Assign roles to an organization
7. Assign roles to an organization
8. Prepare the ITSO B2B CEP store SAR for publishing
9. Create a store archive from the ITSO B2B CEP store template
10. Publish the store from Store Services
11. Copy commerce portal tools JSPs and update properties
12. Verify the ITSO B2B CEP store
13. Deploy the commerce enabled portlets
14. Set commerce portlet permissions
15. Verify the ITSO B2B CEP store portal functionality

Disable WebSphere security before publishing

If WebSphere Portal and WebSphere Commerce are installed on the same node, WebSphere security should be disabled before publishing a store. This is not necessary if they are on separate nodes.

1. Ensure the WebSphere Portal application server is stopped before disabling WebSphere security. Do not start the WebSphere Portal application server until WebSphere security has been enabled after publishing.
2. Start the WebSphere Application Server Administration Console.
3. Click **Console -> Security Center**.
4. From the General tab, clear the **Enable Security** check box.
5. Click **Apply** and then **OK**.
6. Stop and start the WebSphere Administrative Server (IBM WS AdminServer 4.0) for changes to take effect.

After publishing the store, WebSphere security can be enabled.

9.1.1 ITSO sample code

For details on the ITSO sample code, refer to 7.1, "ITSO sample code" on page 286.

9.1.2 Database backup

Back up the WebSphere Commerce instance database and other supporting databases before publishing the ITSO B2B CEP store SAR file.

We recommend a backup of the following databases:

- ▶ WebSphere Commerce <instance> database
- ▶ WebSphere Application Server repository database
- ▶ WebSphere Commerce Payments database

We created a new database backup directory `c:\ibm\ddb\backupstore`.

For details on how to back up a DB2 database, refer to "Back up a DB2 database" on page 375.

9.1.3 Server startup

Ensure the following are started on each of the nodes:

- ▶ WebSphere Portal node:
 - DB2
 - IBM HTTP Server
 - IBM WS AdminServer 4.0
 - WebSphere Portal application server
- ▶ Directory Server node:
 - DB2
 - IBM HTTP Server
 - IBM Directory Server V4.1
- ▶ WebSphere Commerce node:
 - DB2
 - IBM HTTP Server
 - WebSphere Administrative Server (adminserver.bat)
 - WebSphere Commerce Payments application server and IBMPayServer.cmd
 - WebSphere Commerce <instance> application server

9.1.4 Create a new organization for the ITSO B2B CEP store

If you have already published a store for the existing WebSphere Commerce instance, you will need to create a new organization for the ITSO B2B CEP store. If you have not published a store previously, you can use the predefined *Default Organization* as the store owner.

To create a new organization for the ITSO from the WebSphere Commerce Administration Console, do the following:

1. Enter the following URL in a Web browser to start the WebSphere Commerce Administration Console:
`https://<hostname>:8000/adminconsole`
2. Log in as wcsadmin, select **Site**, and click **OK**.
3. Select **Access Management -> Organizations** from the menu bar.
4. In the Organization window, click **New**.
5. Enter the following in the Details window and then click **Next**:
 - Short name (required): CEP
 - Distinguished Name: *Leave this field blank*

Note: When attempting to create an organization with the DN supplied, we received a generic system error. For this reason, we left the Distinguished Name field blank when creating the organization. It will default to the DN configured in LDAP.

- Description: CEP
 - Business category: CEP
 - Organization type: select **Organization**
6. Enter the following in the Address window and then click Finish:
 - Street address (required)
 - City (required)
 - State/Province (required)
 - Zip/Postal code (required)
 - Country/Region (required)
 7. When done, you should see the newly created organization listed. Notice, the parent organization to the CEP organization is Root Organization. Double-click the **CEP** organization. Notice the Distinguished Name is `o=CEP,dc=ibm,dc=com`. Figure 9-1 on page 314 displays the CEP organization from the Administration Console.



Figure 9-1 CEP organization

9.1.5 Create a new user and add to CEP organization

To create a new user as part of the CEP organization, do the following from the WebSphere Commerce Administration Console:

1. From the WebSphere Commerce Administration Console, select **Access Management** -> **Users**.
2. Click **New**.
3. When the Details window appears, we entered the following and then clicked **Next**:
 - Title: (optional)
 - First Name: (optional)
 - Middle Name: (optional)
 - Last Name: buycep1 (required)
 - Logon ID: buycep1 (required)
 - Password: <password> (required)
 - Password confirmation: <password> (required)

- Challenge question: (optional)
 - Answer to challenge question: (optional)
 - Account policy: select **Administrators**
 - Account Status: select **Enabled**
4. When the Business Profile window appears, we entered the following and then clicked **Next**:
- Employee ID: (optional)
 - Employee type: (optional)
 - Department number: (optional)
 - Manager's name: (optional)
 - Administrative assistant's name: (optional)
 - Preferred language: we selected **United States English**
 - Parent organization: click **Find** (required)
5. A listing of the organizations will appear. We checked the **CEP** organization, clicked **Select** and then clicked **Next**.
6. When the Address window appears, we entered the following and then clicked **Next**:
- Street address: 700 Park Office (required)
 - City: RTP (required)
 - State/Province: NC (required)
 - Zip/Postal code: 27709 (required)
 - Country/Region: USA (required)
7. When the Contact window appeared, we did not enter any information since these fields are optional (for test purposes). Click **Finish**.
8. Notice the user we created, `uid=buycepl,o=CEP,dc=ibm,dc=com`, is displayed in the Administration Console as seen in Figure 9-2 on page 316.

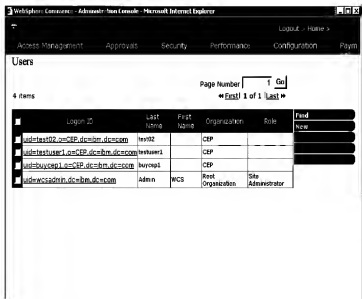


Figure 9-2 WebSphere Commerce Administration Console - testuser1

9.1.6 Assign roles to an organization

In order to assign roles to a user, the organization must first be assigned roles. There are many predefined roles, or roles can be created. In our example, we assigned all possible roles to the CEP organization we created.

To assign the WebSphere Commerce roles to the CEP organization, do the following:

1. From the WebSphere Commerce Administration Console, select **Access Management -> Organizations**.
2. Check the **CEP** organization, and then click **Roles**.
3. Select the desired role and click **Add**. In our example, we clicked **Add All** roles for the CEP organization and then clicked **OK**.

9.1.7 Assign roles to a user of an organization

In our example, we assigned the buyer roles, subset of all roles available, to the buycep1 user of the CEP organization created in 9.1.5, "Create a new user and add to CEP organization" on page 314.

To assign the buycep1 user of the CEP organization buyer roles, do the following:

1. From the WebSphere Commerce Administration Console, select **Access Management** -> **Users**.
2. Check the **uid=buycep1,o=CEP,dc=ibm,dc=com** user.
3. Click **Roles** from the right-hand controls.
4. When the Roles window appears, we selected the following and clicked **Add** for each:
 - Organization: select **CEP**
 - Role: <role> click **Add**

Where <role> is all of the following:

- Procurement Buyer
- Procurement Buyer Administrator
- Buyer (buy-side)
- Buyer (sell-side)
- Buyer Administrator
- Buyer Approver

Note: The roles assigned are for example purposes and may be different for your business needs.

5. When done adding roles, you should see something like Figure 9-3, "Assign roles to buycep1 user" on page 318.



Figure 9-3 Assign roles to buycep1 user

6. After clicking **OK**, you will see the roles assigned to the buycep1 user as displayed in Figure 9-4 on page 319.



Figure 9-4 Roles displayed for the buycep1 user

7. Log out and close the WebSphere Commerce Administration Console.

9.1.8 Prepare the ITSO B2B CEP store SAR for publishing

Before the ITSO B2B CEP store SAR can be published, the WebSphere Commerce node must be prepared as follows:

1. Create a directory in the <WC_HOME>\samplestores directory for the ITSO B2B CEP store (for example, c:\ibm\wc\samplestores\b2bcecp).
2. Copy the b2bCep_en_US_es_ES.sar and Feature_en_US.html files found in the c:\sg246890-code\sar directory to the <WC_HOME>\samplestores\b2bcecp directory.
3. Modify the <WC_HOME>\xml\tools\devtools\SARRegistry.xml file with a text editor to include the entries listed in Example 9-1.

Example 9-1 Sample SARRegistry.xml update for ITSO B2B CEP store

```
<SampleSAR fileName="b2bCep_en_US_es_ES.sar" relativePath="b2bcecp">
```

```
<html locale="en_US" featureFile="b2bcep/Feature_en_US.html"/>
</SampleSAR>
```

4. After you have saved the SARRegistry.xml, do the following for the changes made to take effect:

- Close Store Services

Note: We noticed that the SARRegistry.xml file was locked in some cases even after closing Store Services. In order to save the SARRegistry.xml file without getting a locked file error, sometimes we had to stop the application server, save the SARRegistry.xml, and then start the application server.

9.1.9 Create a store archive from the ITSO B2B CEP store template

To create a store based on the ITSO B2B CEP store template, do the following:

1. Ensure the following are started:

- IBM DB2 services
- IBM HTTP Server (Windows services)
- IBM WS Admin Server 4.0 (Windows services)

If you have modified the adminserver.bat file to include startup parameters, run this instead.

- WebSphere Commerce - <instance_name> (WebSphere application server)
- WebSphere Payment Manager (WebSphere application server)
- IBMPayServer (command executed and running)

2. Start Store Services by entering the following URL from a Web browser:

`https://<hostname>:8000/storeservices`

3. Log on as wcsadmin.

4. Select the sample store that you want your new store to be based on. For example, we selected the following for ITSO and then clicked **OK**:

- Select **b2bCep_en_US_es_ES.sar**
- Store archive: **b2bcep1**
- Store directory: **b2bcep1**
- Store owner: select **CEP**

5. You should see the message **b2bcep1.sar** created successfully. Click **OK**.

You have now created a new store archive named **b2bcep1** based on the **b2bCep_en_US_es_ES.sar** template store.

9.1.10 Publish the store from Store Services

Now that the store archive b2bcep1.sar has been created, do the following to publish the ITSO B2B CEP store from Store Services:

1. To publish the store, check the store you created (for example, we checked b2bcep1.sar) and then click **Publish**.
2. We accepted the defaults for the remaining options.
3. We monitored the <WC_HOME>\instances\<instance>\logs\wcs.log and used the Windows task manager to track publishing progress. Periodically, click the **Refresh** option when CPU activity has declined in the Windows Task Manager. When the publishing status changes to Publishing Completed Successfully, continue to the next step.

If the publishing fails refer to "Troubleshooting a store publishing failure" on page 373.

9.1.11 Copy commerce portal tools JSPs and update properties

This section describes the manual copy of commerce portal tools JSPs as part of the store deployment. The JSP files are needed but not packaged and deployed as part of the SAR file process.

1. Copy:

```
<WAS_HOME>/installedApps/WC_Enterprise_App_instance_name  
.ear/wcstores.war/store_name /CSA/wcstores/portal_jsp/
```

To:

```
<WAS_HOME>/installedApps/WC_Enterprise_App_instance_name.ear/w  
cstores.war/
```

2. Copy:

```
<WAS_HOME>/installedApps/WC_Enterprise_App_instance_name.ear/w  
cstores.war/store_name /CSA/wctools.war/portal_jsp/
```

To:

```
<WAS_HOME>/installedApps/WC_Enterprise_App_instance_name.ear/w  
cstores.war/
```

Copy the contents of the file.

3. In a text editor, open the file
<WAS_HOME>/installedApps/WC_Enterprise_App_instance_name.ear/wcstores.war/store_name/CSA/wcstores/Append_ContractRB_en_US.properties.
Add the content of the Append_ContractRB_en_US.properties file to the end of the ContractRB_en_US.properties file.

4. In a text editor, open the file
<WAS_HOME>/installedApps/WC_Enterprise_App_instance_name.ear/properties/com.ibm.commerce/tools/contract/properties/ContractRB_en_US.properties.

Add the content of the Append_ContractRB_en_US.properties file to the end of the ContractRB_en_US.properties file.

5. In a text editor, open the file
<WAS_HOME>/installedApps/WC_Enterprise_App_instance_name.ear/wcsources.war/store_name/CSA/wctools/Append_OrderLabels_en_US.properties.

Copy the contents of the file.

6. In a text editor, open the file
<WAS_HOME>/installedApps/WC_Enterprise_App_instance_name.ear/properties/com.ibm.commerce/tools/order/properties/OrderLabels_en_US.properties.

Add the content of the Append_OrderLabels_en_US.properties file to the end of the OrderLabels_en_US.properties file.

7. From a DB2 command window, connect to your WebSphere Commerce instance database and enter the following:

```
insert into cntrdisply (devicefmt_id,cntrdisply_id,displaypagename) values  
(-4,1,'tools/contract/ContractSummary.jsp')
```

Note: The -4 value above is the default value for devicefmt_id of the PortalBrowserAdapter (specified in the instance_name.xml). If you have changed this value in the instance_name.xml file, that value should be used instead of -4.

Refer to 5.2.13, "Enable WebSphere Commerce portal adapter" on page 196 for details on how to check the values in the database and update the WebSphere Commerce instance XML file accordingly.

8. Restart the WebSphere Commerce instance from the WebSphere Application Server Administration Console.

9.1.12 Verify the ITSO B2B CEP store

After publishing the store, we recommend that you verify the functionality of the standard WebSphere Commerce JSPs by directly accessing the WebSphere Commerce node, and the functionality of the commerce enabled portal store.

1. To perform a basic verification test of the store after publishing is finished, click **Publish Summary -> Launch Store**. Add the store to your favorites or bookmarks list.

Alternatively, enter the store URL such as:

`http://<wc_hostname>/webapp/wcs/stores/servlet/b2bcep1/index.jsp`

2. Register a new user.
3. Log on with the new user.
4. Browse the store catalog.
5. Select an item to purchase and add to shopping cart.
6. Complete an order by selecting a valid payment type. For example, we selected Visa and entered the 16 zeroes (0) as the card number for test purposes (or 4111111111111111, 15 zeros).
7. Verify the order from WebSphere Commerce Payments.
 - a. Log on to WebSphere Commerce Payments:
`http://<wcpay_hostname>/webapp/PaymentManager`
 - b. Click **Approve**. You should see the order that was just placed. Select the order and click **Approve**.
 - c. You should see a message indicating the order was approved and that there are no more pending orders to approve.

9.1.13 Deploy the commerce enabled portlets

This section describes how to deploy the commerce enabled portlets supplied with the B2B Direct sample of the IBM Commerce Enhancement Pack - April 2003 Edition as a base for the ITSO B2B CEP store sample.

Note: For more information, refer to *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*.

To deploy and configure the commerce enabled portal runtime for the B2B Direct sample, do the following on WebSphere Portal node:

1. Ensure you have published one of the following commerce enabled portal store SAR files:

Note: This step will have already been completed if you followed the procedure documented 9.1, "Deploy the ITSO B2B CEP store to the runtime environment" on page 310.

- ITSO B2B CEP store SAR by following the steps in:
 - 9.1.2, "Database backup" on page 311
 - 9.1.4, "Create a new organization for the ITSO B2B CEP store" on page 312
 - 9.1.8, "Prepare the ITSO B2B CEP store SAR for publishing" on page 319
 - 9.1.9, "Create a store archive from the ITSO B2B CEP store template" on page 320
 - 9.1.10, "Publish the store from Store Services" on page 321
 - 9.1.15, "Verify the ITSO B2B CEP store portal functionality" on page 331

Or

- B2B Direct sample SAR (refer to *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition*)
2. Change to the <CEP_HOME>\B2BDirectPortal\CommerceRefApp\bin directory.

Note: This directory should include the files from the IBM Commerce Enhancement Pack - April 2003 Edition.

3. Rename the WebsphereCommerceEnabledPortalB2BDirect.war file to WebsphereCommerceEnabledPortalB2BDirect_original.war.

This has the effect of creating a backup, since we will modify files and repackage the WAR file as WebsphereCommerceEnabledPortalB2BDirect.war.

4. Set PATH to include Java:

```
set PATH=%PATH%;c:\ibm\was\java\bin
```

5. Unpack the `WebSphereCommerceEnabledPortalB2BDirect_original.war` file:

```
jar -xvf WebSphereCommerceEnabledPortalB2BDirect_original.war
```

6. Edit the `WEB-INF/portlet.xml` file:

- a. Search for all `hostname.domain` entries and replace them with your WebSphere Commerce node host name and domain (for example, `wcserv1.itso.ral.ibm.com`).
- b. In our scenario, our environment is configured to use SSL. Change all portlet URLs in the `portlet.xml` accessing port 8000 from `http://` to `https://` in the `portlet.xml` file. For example, in our environment we replaced all occurrences as follows:

```
http://wcserv1.itso.ral.ibm.com:8000/  
https://wcserv1.itso.ral.ibm.com:8000/
```

- c. Update the values of for the following parameters in the `portlet.xml` file to match your published store.
 - `storeId`
 - `catalogId`
 - `langId`
 - `categoryId`
 - `productId`
 - `parent_category_rn`

Table 9-1 on page 326, includes the values entered for ITSO B2B CEP store. The values for your store may be different depending on whether you have published stores prior to the ITSO B2B CEP store. Also, as you add/delete products and categories, the reference ID numbers will change.

Note: When updating the information in the `portlet.xml` file, we have the following recommendations for updating these values:

- ▶ We suggest that you update the parameters in the order listed.
- ▶ The `storeId`, `catalogId` and `langId` are the easiest to find and update.
- ▶ Ensure you update the value only.
- ▶ When updating the `categoryId` and `productId` parameters, please note that these values may change as catalog data is resolved and loaded. You may have to update the `portlet.xml` file in the future as the resolved ID numbers change. Find the `categoryId` values for the following used in the B2B commerce enabled portal sample. For example:
 - Woodworking saws
 - Sanders
 - Lathes
 - Drills
 - Grinders
 - Screwdrivers
- ▶ For test/development purposes, we chose some categoryIds and updated one productId value for them and the `parent_category_rn`.
- ▶ The highest level category is the `catalogId` (`parent_category_rn` for high-level categories).

There are two methods than can be used to obtain the values of the parameters (database columns):

- ▶ Record values from WebSphere Commerce URL when navigating the site categories and products.
- ▶ Perform DB2 SQL queries on the WebSphere Commerce instance database. For example:

```
db2 connect to <wc_database>
db2 select * from CATGROUP
```

Table 9-1 ITSO B2B CEP store `portlet.xml` values

Parameter/element	Value
<code>hostname.domain</code>	<code>wcserv1.itso.ral.ibm.com</code>
<code>langId</code>	<code>-1</code>
<code>storeId</code>	<code>10001</code>

Parameter/element	Value
catalogId	10001
Category portlet updates	
Woodworking saws categoryId * parent_category_m Note: The parent category for the highest level categories is the catalogId.	10005 10001
Sanders categoryId * parent_category_m	10006 10001
Lathes categoryId * parent_category_m	10007 10001
Drills categoryId * parent_category_m	10008 10001
Grinders categoryId * parent_category_m	10009 10001
Screwdrivers categoryId * parent_category_m	10010 10001
Airtools categoryId * parent_category_m	10011 10001
Cordless saws categoryId * parent_category_m	10012 10001
Handsaws categoryId * parent_category_m	10015 10001
Drill bits categoryId * parent_category_m	10013 10001
Chargers categoryId * parent_category_m	10014 10001
Product portlet updates	
Circular saw productId * Woodworking parent_category_m	10001 10005 (Woodworking saw categoryId)
Electric Sander productId Sanders parent_category_m	10037 10006 (Sander categoryId)

Parameter/element	Value
Second Operation Lathe productId * Lathes parent_category_m	10073 10007 (Lathe categoryId)
Hammer Drill productId * Drills parent_category_m	10109 10008 (Drill categoryId)
Small Angle Grinder productId * Grinders parent_category_m	10154 10009 (Grinder categoryId)
Power Screwdrivers productId * Drywall Screwdrivers parent_category_m	10190 10010 (Screwdriver categoryId)

7. Now that we have completed the modifications to the portlet.xml file, the WebsphereCommerceEnabledPortalB2BDirect.war file needs to be repackaged, as follows:
 - a. Change to the <CEP_HOME>\B2BDirectPortal\CommerceRefApp\bin directory.
 - b. Set PATH to include Java:


```
set PATH=%PATH%;c:\ibm\was\java\bin
```
 - c. Repackage the WAR file to include the updates:


```
jar -cvf WebsphereCommerceEnabledPortalB2BDirect.war WEB-INF
```
8. Deploy the updated B2B Direct sample portlets, as follows:
 - a. Change to the <CEP_HOME>\B2BDirectPortal\CommerceRefApp\scripts directory.
 - b. Run the following command to deploy the B2B direct sample portlets:


```
installb2bdirectcommerceportal.bat
```

You will be prompted with the following (for our example, we entered the following values):

 - WPS Root: c:\ibm\PortalServer
 - WPS Admin UID: wpsadmin
 - WPS Admin Password: wpsadmin
 - WPS Access URL: http://wcportal1.itso.ra1.ibm.com/wps

Note: After the WebsphereCommerceEnabledPortalB2BDirect.war is deployed, the deployed version of the portlet.xml file can be found in the <WAS_HOME>\InstalledApps\WebSphereCommerceEnabledPortalB2BDirect\stamp>\WebSphereCommerceEnabledPortalB2BDirect.war\WEB-INF directory. For test purposes, you may edit the portlet.xml file from this location. For production deployment and proper testing, we recommend that you edit portlet.xml, repackage the WAR file and then redeploy it.

Personalization (optional)

If you choose to set up personalization, refer to the *Commerce Enabled Portal Integration Guide, IBM Commerce Enhancement Pack April 2003 Edition* on deploying the personalization workspace and associated scripts.

9.1.14 Set commerce portlet permissions

After the WebSphere Portal installation and configuration, set the commerce portlet permissions within the WebSphere Portal Administration Console. Log in to the portal as `wpsadmin` and add at least View permission for Commerce Enhancement Pack portlet application, portlets, places and pages for portal users/groups.

1. From a Web browser, enter the following URL:
`http://<portalserver_fullyqualified_hostname>/wps/portal`
2. In the upper right, click **Log in**. You will be prompted for the following, and then click **Log in**.
 - User ID: `wpsadmin`
 - Password: `wpsadmin`
3. Select the **Portal Administration** tab.
4. Select the **Access Control List**, under Security.
5. Set permission for portlets.

Note: For test purposes, we set permissions for all authenticated users to have access. In a production environment, you will need to set permissions for specific groups or users.

- a. Select the **Special Groups** radio button. From the pull-down, select **All Authenticated Users**.
- b. Select the objects for permissions. In this case, we selected **portlets**.
- c. Select the **Show All** radio button.
- d. Click **Go** to display the list of portlets.
- e. Check the view permissions for the following commerce enabled portlets, then click **Save**:
 - Account List
 - Advanced Search
 - Catalog
 - Contract List
 - Current Orders
 - Drills

- Drills SKUs
- Grinders
- Grinder SKUs
- Invoice List
- Lathes
- Lathes SKUs
- My Account
- MyAlert
- Order Approval
- Order History
- Order List
- Product Catalog
- Product Search
- RFQ Approval
- RFQ List and Launch RFQ tool
- RFQ Response List
- Requisition List
- Sanders
- Sanders SKUs
- Screwdrivers
- Screwdrivers SKUs
- SendAlertMessage
- Woodworking Saws
- Woodworking Saws SKUs
- myCatalog
- quickOrder

6. Set permissions for the portlet application:

- a. From the Select the objects for the permissions pull-down, select **Portlet applications**.
- b. Click **Go**.
- c. From the list of portlet applications, check the view permission for the **Commerce Portal Application** and then click **Save**.

7. Set permissions for pages and places.

Note: The WebSphere Portal permissions only need to be set manually if you are not using personalization. The replacement **Login** command provided with CEP will override all portal access control lists according to personalization rules.

- a. From the Select the objects for the permissions pull-down, select **Pages**.
- b. Click **Go**.

- c. From the list of the following *places* and corresponding *pages*, check the view or manage permissions (as appropriate for your needs) and then click **Save**:
 - Buyer Organization
 - MyStore
 - Seller Organization
8. Log off the WebSphere Portal admin user (wpsadmin).

9.1.15 Verify the ITSO B2B CEP store portal functionality

Now that the runtime is configured and the ITSO B2B CEP store sample has been deployed, we need to verify the functionality of the B2B commerce enabled portal application. If SSO is working, a new token cookie is received by the browser after a successful connection. After logging on to the WebSphere Portal, we will access a WebSphere Commerce page and should not be prompted to log on.

To verify the B2B commerce enabled portal application, do the following:

1. Open a Web browser and type the URL:

`http://<portal_fully_qualified_hostname>/wps/portal`

Note: Ensure you have entered the fully qualified host name.

2. Click **Log in** found in the upper-right of the page. A login window is displayed. Enter the following and then click **Submit Login**.

- Account: wcsadmin
- Password: <wcsadmin_password>

Note: The WebSphere Commerce admin user is wcsadmin.

3. Click **MyStore**.

You should see the WebSphere Portal welcome page (see Figure 9-5 on page 332).

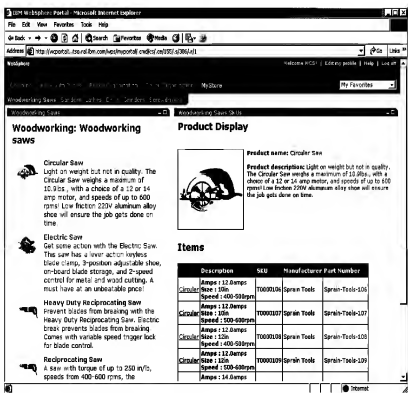


Figure 9-5 ITSO B2B CEP store MyStore home page

- Experiment with the other options listed.

Congratulations! You have completed the ITSO B2B CEP store deployment to the WebSphere Commerce Portal runtime environment.

9.2 Deploy the ITSO B2B CEP store to the development test environment

After completing the configuration of development environment, as described in Chapter 6, "Implement the development environment" on page 253, the ITSO B2B CEP store can be deployed to the development environment.

The high-level steps to deploy the ITSO B2B CEP store are as follows:

1. ITSO sample code
2. Database backup
3. Create CEP organization
4. Prepare the ITSO B2B CEP store template for publishing
5. Create the CEP store from the ITSO B2B CEP store template
6. Publish the CEP store
7. Post-publish CEP store configuration
8. Deploy the commerce enabled portlets
9. Verify commerce enabled portal

9.2.1 ITSO sample code

For details on the ITSO sample code refer to 9.1.1, "ITSO sample code" on page 311.

9.2.2 Database backup

Before you deploy the ITSO B2B CEP store in the development environment, we recommend that you back up the WebSphere Commerce instance database.

Refer to "Back up a DB2 database" on page 375 for details.

9.2.3 Create CEP organization

Create a new organization for the fictitious working example CEP store from the WebSphere Commerce Administration Console.

1. Open WebSphere Studio Application Developer by Start -> **Programs -> IBM WebSphere Studio -> Application Developer 5.0.**
2. Open the **Server** perspective. In the Servers window, right-click the **WebSphereCommerceServer** server and select **Start**.
3. Enter the following URL in a Web browser to start the WebSphere Commerce Administration Console in the WebSphere Test Environment:

`http://localhost:8080/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=adminconsole.AdminConsoleLogon`

Note: Make sure you use external Microsoft Internet Explorer 5.5 or above. Do not use WebSphere Studio Application Developer integrated browser. If you want to open the browser from WebSphere Studio Application Developer, then change the WebSphere Studio Application Developer Web browser preference to use external browser options.

Refer to "Configure Web browser for WebSphere Studio Application Developer" on page 263 for details.

4. Log in as wcsadmin, select **Site** and click **OK**.
5. Select **Access Management -> Organizations** from the menu bar.
6. From the Organization window, click **New**.
7. Enter the following in the Details window and then click **Next**:
 - Short name (required): CEP
 - Distinguished Name: o=CEP,o=Root Organization
 - Description: CEP
 - Business category: CEP
 - Organization type: select **Organization**
8. Enter the following in the Address window and then click **Finish**:
 - Street address (required):
 - City (required):
 - State/Province (required):
 - Zip/Postal code (required):
 - Country/Region (required):
9. Log out and close the WebSphere Commerce Administration Console.

9.2.4 Prepare the ITSO B2B CEP store template for publishing

Prior to publishing the ITSO B2B CEP store, we need to create a new directory for the ITSO B2B CEP store template SAR file, configure the SARRegistry.xml for the template to be visible from Store Services, and create a new store for the CEP organization.

1. Create a directory for the ITSO B2B CEP store template on the WebSphere Commerce Portal development node in the <CEP-HOME>\samplestores directory. For example, <CEP-HOME>\samplestores\cep, where <CEP-HOME> is the physical path where you installed the WebSphere Commerce Enhancement Pack.
2. Copy the ITSO provided B2B CEP sample store template SAR file, c:\sg246890-code\sar\b2bCep_en_US_es_ES.sar and Features_en_US.html from the c:\sg246890-code\sar\ directory to the

<CEP-HOME>\samplestores\cep\ directory on the WebSphere Commerce Portal development node:

3. Update the SARRegistry.xml to add the ITSO B2B CEP store template.

Edit the <CEP-HOME>\xml\tools\devtools\SARRegistry.xml to include the ITSO B2B CEP store template SAR elements as seen in Example 9-2.

Example 9-2 Sample B2B CEP SAR elements for SARRegistry.xml

```
<SampleSAR fileName="b2bCep_en_US_es_ES.sar" relativePath="cep">
  <html locale="en_US" featureFile="cep/Feature_en_US.html" />
</SampleSAR>
```

Note: In the event that Store Services was started prior to updating the SARRegistry.xml, restart Store Services for the entry to become available. It is not necessary to stop/start the application server.

4. Make sure you have created a new organization for the fictitious working example CEP store from the WebSphere Commerce Administration Console. If you did not create the CEP organization, then follow the steps in 9.2.3, "Create CEP organization" on page 333.

9.2.5 Create the CEP store from the ITSO B2B CEP store template

To create a store based on the ITSO-provided B2B CEP store template from within Store Services, do the following:

1. Open WebSphere Studio Application Developer by Start -> Programs -> IBM WebSphere Studio -> Application Developer 5.0.
2. Open the Server perspective. In the Servers window, right-click **WebSphereCommerceServer** and select **Start**.

Note: After logging on to Store Services, you will have to wait until the page is compiled before it is displayed (can take a couple of minutes). Use the Web browser Refresh option.

3. Start Store Services by entering the following URL from a Web browser:

```
http://localhost:8080/webapp/wcs/tools/servlet/ToolsLogon?XMLFile=devtools.
Logon
```

Note: Make sure you use external Microsoft Internet Explorer 5.5 or above. Do not use the WebSphere Studio Application Developer integrated browser. If you want to open the browser from WebSphere Studio Application Developer, then change the WebSphere Studio Application Developer Web browser preference to use external browser options.

Refer to "Configure Web browser for WebSphere Studio Application Developer" on page 263 for details.

4. Log on using the user ID `wcsadmin` and `<password>`.

Tip: If your account becomes disabled, refer to "Reset a disabled account" on page 376.

5. The Store Archives page will open if you already have a store published. Click **New** to go to the Create Store Archive page. If there are no published stores, Create Store Archive page will be displayed automatically.
6. From Store Services, select the sample store that you want your new store to be based on. For example, we selected the following for the CEP and then clicked **OK**:
 - Select **b2bCep_en_US_es_ES.sar**
 - Store archive: `b2bCep1`
 - Store directory: `b2bCep1`
 - Store owner: select **CEP**
7. You should see the message `b2bCep1.sar created successfully`. Click **OK**.

You have now created a new store archive named `b2bCep1` based on the `b2bCep_en_US_es_ES.sar` template store. Do not close Store Services, since we will use it to publish the store in next step.

9.2.6 Publish the CEP store

Now that the store archive for the `b2bCep1` has been created, do the following to publish the ITSO B2B CEP store from Store Services:

1. To publish the store, check the store you created (for example, we checked `b2bCep1.sar`) and then click **Publish**.
2. We accepted the defaults for the remaining options.
3. To review the publishing logs in the event of a failure, check the following:

`<CEP-HOME>\instances\<instance>\logs\messages.txt`

```
<CEP-HOME>\instances\<instance>\logs\trace  
<CEP-HOME>\instances\<instance>\logs\ecmsg_hostname_timestamp.log  
<CEP-HOME>\instances\<instance>\logs\wcs.log
```

In the event of a publishing failure: On more than one occasion, the status did not change from *publishing* to *published*. We have found using the Windows Task Manager to monitor CPU utilization useful.

First, try closing Store Services after waiting and verifying from the task manager that CPU activity is low. Restart Store Services and check the publishing status. Sometimes it really is completed, while at other times you will need to stop the application server and republish.

4. To perform a basic verification test of the store after publishing is finished, click **Publish Summary** -> **Launch Store**. Add the store to your favorites or bookmarks list. The published store also can be open by entering the following URL:

```
http://localhost:8080/webapp/wcs/stores/servlet/Logout?storeId=10051&langId=-1
```

Enter the values of the storeId and langId for your store, which might be different from the above-mentioned.

9.2.7 Post-publish CEP store configuration

After you have published the store using Store Services, you must manually copy the following files and directories:

1. Copy the portal_jsp folder and subfolders from
<CEP-WORKSPACE-HOME>\wcstores\Web Content\<store_name>\CSA\wcstores\portal_jsp\ (for the ITSO SAR file, <store_name> will be replaced by b2bCep1) to
<CEP-WORKSPACE-HOME>\wcstores\Web Content\.
2. Copy portal_jsp folder and subfolders from
<CEP-WORKSPACE-HOME>\wcstores\Web Content\<store_name>\CSA\wctools\portal_jsp\ (where the ITSO SAR file <store_name> will be replaced by b2bCep1) to
<CEP-WORKSPACE-HOME>\wctools\Web Content\.
3. In a text editor, open the following two files:
 - <CEP-WORKSPACE-HOME>\WebSphereCommerceServer\properties\com\ibm\commerce\tools\contract\properties\ContractRB_en_US.properties
 - <CEP-WORKSPACE-HOME>\wcstores\Web Content\bebCep1\CSA\wctools\Append_ContractRB_en_US.properties

Add the contents of the Append_ContractRB_en_US.properties file to the end of the ContractRB_en_US.properties file. Save the file.

4. Save the ContractRB_en_US.properties file.
5. In a text editor, open the following two files:
 - <CEP-WORKSPACE-HOME>\wcstores\WebContent\bebCep1\CSA\wc\tools\Append_OrderLabels_en_US.properties
 - <CEP-WORKSPACE-HOME>\WebSphereCommerceServer\properties\com\ibm\commerce\tools\order\properties\OrderLabels_en_US.properties

Add the contents of the Append_OrderLabels_en_US.properties to the end of the OrderLabels_en_US.properties file. Save the file.

9.2.8 Deploy the commerce enabled portlets

To configure the commerce enabled portal runtime for the B2B Direct sample, do the following:

1. Modify the <instance>.xml file, which is located in
<CEP-HOME>\instances\<instance-name>\xml\.

For example, in ITSO the full path for the <instance>.xml is
c:\ibm\cep\instances\wcdev\xml\wcdev.xml.

- a. Set the appropriate HTTP Adapter device Format Id for Portal. For details see in 5.2.13, "Enable WebSphere Commerce portal adapter" on page 196.
- b. Set the appropriate HTTP Adapter device Format Id for WAP Portal. For details see in 10.2, "Deploying the WAP portal adapter" on page 350.

Note: This step is only required if you want to use WAP Portal development.

- c. Set cache daemon to false (see Example 9-3).

Example 9-3 No cache setup

```
<component compClassName="com.ibm.commerce.cache.daemon.CacheDaemonComponent"
  enable="false"
  name="CacheDaemon" />
```

2. Creating the wpsdebug user includes selecting the organization and user role and set using the WebSphere Commerce Administration Console. Refer to the following for detailed instructions:
 - a. 9.1.5, "Create a new user and add to CEP organization" on page 314.
 - b. 9.1.6, "Assign roles to an organization" on page 316.

- c. 9.1.7, "Assign roles to a user of an organization" on page 317.

Note: The `wsdebug` user is required for the ITS0-provided debug capability between the portal and commerce. For test purposes, we assigned the site administrator role to `wsdebug` and all roles to the CEP organization.

Update B2B store portlet application

From now on, the rest of the steps will be done on WebSphere Portal application server (in our example, single-node development environment).

3. Change to the `C:\temp\CEP\B2BDirectPortal\CommerceRefApp\bin` directory.

4. Rename the `WebSphereCommerceEnabledPortalB2BDirect.war` file to `WebSphereCommerceEnabledPortalB2BDirect_original.war`.

This has the effect of creating a backup, since we will modify files and repackage as `WebSphereCommerceEnabledPortalB2BDirect.war`.

5. Set `PATH` to include Java:

```
set PATH=%PATH%;%WSAD-HOME%\runtimes\aes_v4\java\bin
(example in ITS0, set PATH=%PATH%;c:\ibm\wsad\runtimes\aes_v4\bin)
```

6. Unpack the `WebSphereCommerceEnabledPortalB2BDirect_original.war` file:

```
jar -xvf WebSphereCommerceEnabledPortalB2BDirect_original.war
```

7. Edit the `WEB-INF/portlet.xml` file:

- a. Replace all `hostname.domain` entries with your WebSphere Commerce node host name and domain. For example, in ITS0:

```
<hostname>:8080
```

- b. Update the values for the following parameters in the `portlet.xml` file to match your published store:

- `storeId`
- `catalogId`
- `langId`
- `categoryId`
- `productId`
- `parent_category_rn`

Note: When updating the information in the `portlet.xml` file, we have the following recommendations, although we are not aware of a painless method of updating these values.

- ▶ We suggest that you update the parameters in the order listed.
- ▶ The `storeId`, `catalogId`, and `langId` are the easiest to find and update.
- ▶ Ensure you update the value only.
- ▶ When updating the `categoryId` and `productId`, please note that these values may change as catalog data is resolved and loaded. You may have to update the `portlet.xml` file in the future as the resolved ID numbers change. Find the `categoryId` values for the following used in the B2B commerce enabled portal sample:
 - Woodworking saws
 - Sanders
 - Lathes
 - Drills
 - Grinders
 - Screwdrivers
- ▶ For development purposes, we chose some categories and updated one `productId` value for them and the `parent_category_rn`.
- ▶ The highest level category is the `catalogId` (`parent_category_rn` for high-level categories).

There are two methods that can be used to obtain the values of the parameters (database columns):

- ▶ Record values from WebSphere Commerce URL when navigating the site categories and products.
 - ▶ Perform DB2 SQL queries on the WebSphere Commerce instance database.
- c. In a development environment, we need to turn off `SSOEnabled` to access WebSphere Commerce from the portlet. Search for `<param-name>SSOEnabled</param-name>` in the `portlet.xml` file and set *false* to param value. For example, in our environment we replaced all occurrences, as seen in Example 9-4.

Example 9-4 Update SSOEnabled attributes in portlet.xml

```
<config-param>
  <param-name>SSOEnabled</param-name>
  <param-value>>false</param-value>
```

- d. In a development environment, replace all 8000 port number with 8080 in the portlet.xml file. For example, in our environment we replaced all occurrences as follows:

```
hostname.domain:8000
<hostname>:8080
```

8. Delete the commerceportal.jar file from the C:\temp\cep\B2BDirectPortal\CommerceRefApp\bin\WEB-INF\lib\ directory, since we are going to replace it with the ITSO-modified version, so that even though SSO is not found in c:\sg246890-code\debug, we still can access WebSphere Commerce from WebSphere Portal.

Important: The ITSO-modified commerceportal.jar file should only be used for debug purposes and is not supported by WebSphere Commerce.

Do not use this file in production.

9. Copy the c:\sg246890-code\debug\WebsphereCommerceBasePortlet.jar file, which is included with this redbook sample code, into the C:\temp\cep\B2BDirectPortal\CommerceRefApp\bin\WEB-INF\lib\ directory and rename to commerceportal.jar.
10. To configure a WebSphere Commerce logon URL and password for commerce portlet, do the following:

Note: For details see "Configuring WebSphere Commerce logon URL and password" on page 368.

- a. Copy the c:\sg246890-code\debug\WCSLogon.properties file to the <WAS_HOME>\lib\app\config directory.
 - b. Update the WCSLogon.properties file as described in "Configuring WebSphere Commerce logon URL and password" on page 368.
 - c. On WebSphere Commerce development node, create an empty file called dummy.html in the <CEP-WORKSPACE-HOME>\wcstores\Web Content directory.
11. Now that we have completed the modifications to the portlet.xml file, the WebsphereCommerceEnabledPortalB2BDirect.war file needs to be repackaged.

- a. Change to the C:\temp\CEP\B2BDirectPortal\CommerceRefApp\bin directory.
- b. Set PATH to include Java:

```
set PATH=%PATH%;%MSAD-HOME%\runtimes\aes_v4\bin
```


(example in ITS0, set PATH=%PATH%;c:\ibm\wsad\runtimes\aes_v4\java\bin)
- c. Repackage the WAR file to include the updates as follows:

```
jar -cvf WebsphereCommerceEnabledPortalB2BDirect.war WEB-INF
```

Deploy commerce portlets to WebSphere Studio Application Developer test environment

To deploy the commerce portlets to the WebSphere Studio Application Developer test environment, do the following:

1. From the WebSphere Studio Application Developer workspace, open the Portlet perspective.
2. Right-click in the J2EE Navigator pane, then select **New -> Project**.
3. In the next window, select **Portlet Development** and **Portlet Application Project**. Click **Next**.
4. Enter the Project name (for example, B2B Store Portlets). For Enterprise Application Project, select **New** and enter the new project name (for example, B2BPortletEAR). Leave other fields at the default settings as shown in Figure 9-6 on page 343. Click **Next**.

Create a Portlet Project

Define the Portlet Project

Create a sample portlet application project.

Project name: B2B Store Portlets

☒ Use default:

New project location: C:\Documents and Settings\saumanis\My Documents\IBM\wscd\w... Browse...

Enterprise application project: ☒ New ☐ Existing

New project name: B2BPortletEAR

☒ Use default:

New project location: C:\Documents and Settings\saumanis\My Documents\IBM\wscd\... Browse...

Context root: B2B Store Portlets

< Back Next > Finish Cancel

Figure 9-6 Create a portlet project

5. Select Portlet type: **None**. Click **Finish**.
6. In J2EE Navigator pane, right-click the **B2B Store Portlets** project. Select **Import**.
7. In the next window, select **WAR** file. Click **Next**.
8. In WAR file: field, enter full path to the `WebSphereCommerceEnabledPortalB2BDirect.war` portlet application WAR file that was created in "Update B2B store portlet application" on page 339, or click **Browse** to select the WAR file.
9. For the Web project, select **Existing**. Next to the Existing project name field, click **Browse** and select **B2B Store Portlets**. Click **OK**.
10. Select the **Overwrite existing resources without warning** check box. The window should look similar to Figure 9-7 on page 344. Click **Finish**.

Import Resources from a WAR File

Identify the WAR File and Import Options

Import resources from a WAR file.

WAR file: C:\temp\CEP\WebSphereCommerceEnabledPortals\B2BDirect.war Browse...

Where do you want the imported resources to go?

Web project: ☐ New ☒ Existing

Existing project name: B2B Store Portlets Browse...

Context Root: /B2B Store Portlets

Enterprise application project: ☐ New ☒ Existing

Existing project name: B2BPortletEAR Browse...

Options:

☒ Overwrite existing resources without warning

< Back Next > Finish Cancel

Figure 9-7 Import portlets from a WAR file

11. In J2EE Navigator pane, right-click the **B2B Store Portlets** project and select **Rebuild Project**. After rebuilding completes, there may still be four warnings in the Tasks pane. This is normal.
12. In the Server Configuration pane, right-click **WP 4.2.1 Test Server**. Select **Add -> B2BPortletEAR**.
13. Restart WP 4.2.1 Test Server.

9.2.9 Verify commerce enabled portal

To run the portlets, right-click **B2B Store Portlets** in J2EE Navigator and select **Run on Server**.

In the WP Test Environment server, all portlets are rendered on the same portal page. To improve debugging performance, we recommend running only those portlets that will be rendered on the same page. To disable unused portlets, open the `portlet.xml` file in WebSphere Studio Application Developer and remove Portlet Application entries for the unused portlets.

Note: To debug the entire portlet application with multiple pages, you will need to use WebSphere Studio Application Developer with WebSphere Portal installed on a remote WebSphere Application Server Single Server Edition. This scenario is described in *Integrating WebSphere Commerce V5.4 and WebSphere Portal V4.1.4*, REDP3684.

Congratulations! You have now configured end-to-end WebSphere Commerce and the WebSphere Portal development environment, and deployed the ITSO B2B CEP store.



Creating a commerce enabled portal for mobile clients

This chapter describes how to enable the commerce portal for mobile clients such as WAP mobile devices. In addition, we provide information on the WAP toolkits (simulators) and explain the modifications made to the WAP WML JSPs provided in the ITSO sample code workspace and ITSO B2B CEP store SAR.

The chapter is organized into the following sections:

- ▶ Introduction
- ▶ Deploying the WAP portal adapter
- ▶ OpenWave.SDK 6.2 for WAP toolkit and simulator
- ▶ Sample WAP WML commerce portlet JSPs

10.1 Introduction

In recent times, there has been a dramatic increase in the use of powerful mobile devices. These mobile devices, ranging from pagers to mobile phones, wireless PDAs, and wireless laptops, are changing the way people interact at work and at home. The sophistication of the mobile devices and wireless technologies has advanced to the stage of mass-market usage and acceptance. Just as we saw a surge in e-commerce by PC browser clients, we are now experiencing a similar phenomenon in mobile commerce by users of these more advanced mobile devices.

The combination of wireless technologies and the powerful Java-based application architecture provided by WebSphere Portal and WebSphere Commerce promises to transform e-business. IBM is leading the way in the mobile computing arena by bringing together the technologies of the wireless world and the strength and integration of the IBM server and software family of products, to create end-to-end mobile commerce enabled portal solutions.

Mobile computing, also known as pervasive computing, provides a series of technologies that enable people to complete personal and professional tasks using this new class of portable, intelligent, wireless mobile devices. These mobile devices give people access to information at any time and place. Some areas of the world are more invested in wireless technologies than others. However, the capability and number of users in all areas of the world are growing at a tremendous rate.

Wherever you live, mobile computing will in the very near future become a prominent means of accessing information on the Internet. Just as the PC browser client market matured from accessing the Internet simply to browse and gather information to full-blown e-business and e-commerce, so the same is happening for mobile devices.

As a leader in e-business and e-commerce, IBM has recognized that mobile computing offers tremendous business opportunities, has integrated mobile device support into its products, and has developed a series of middleware products designed as infrastructure for mobile computing.

IBM provides a wide range of electronic commerce solutions for your business needs. Within the IBM WebSphere brand, IBM has enabled the WebSphere Portal and WebSphere Commerce product suites to support mobile clients:

- ▶ IBM WebSphere Commerce

This product suite is very versatile and can be used for B2C, B2B, and Auction e-commerce Web sites. WebSphere Commerce provides a pure Java

programming model that is conducive to supporting mobile clients. Features have been added to provide integration and support for mobile devices.

- IBM WebSphere Portal

This product suite is very versatile and can be used for any kind of portal Web site. WebSphere Portal provides a pure Java programming model that is conducive to supporting mobile clients. Features have been added to provide integration and support for mobile devices.

In today's fast-moving business world, it is important to keep up with the ever-changing demands of users. Computing is no longer confined to desktop PCs and people now expect to be able to access information at any time and from any place with the device they happen to be using at the time. Accordingly, the applications in this environment are varied and encompass everything from simple Web browsing to secure access to business-critical data. There are, therefore, many challenges to be addressed in this environment, and the Commerce Enhancement Pack provides tools that address these issues.

10.1.1 WAP

The Wireless Application Protocol (WAP) is a standard for providing cellular phones, pagers, and other handheld devices with secure access to e-mail and text-based Web pages. Introduced in 1997 by Phone.com, Ericsson, Motorola and Nokia, WAP provides a complete environment for wireless applications that includes a wireless counterpart of TCP/IP and a framework for telephony integration, such as call control and phone book access.

WAP features the Wireless Markup Language (WML), which was derived from Phone.com's HDML and is a streamlined version of HTML for small-screen displays. It also uses WMLScript, a compact JavaScript-like language that runs in limited memory. WAP also supports handheld input methods such as a keypad and voice recognition. Independent of the air interface, WAP runs over all the major wireless networks in place now and in the future. It is also device-independent, requiring only minimum functionality in the unit so that it can be used with a wide range of different phones and handheld devices.

WAP was developed because of the strong limitations of both mobile devices and wireless networks. Most wireless devices are often very limited in terms of display, processing power and available memory. Moreover, wireless networks themselves are characterized by limited bandwidth and high latency. In order for wireless devices to be able to access Internet content in a way similar to wireline PC browser clients, WAP was developed. The WAP specification defines an architecture that optimizes the use of the radio resource and also minimizes the capabilities required for the device.

The main elements of the WAP specification are as follows:

- ▶ A Wireless Application Environment (WAE), which includes a microbrowser and is very well suited for devices with poor capabilities.
- ▶ A new markup language called the Wireless Markup Language (WML), and a new scripting language called WMLScript.
- ▶ A new protocol stack, which is independent of the underlying network and is suitable for connections that have low bandwidth and are often unreliable.
- ▶ A secure protocol called Wireless Transport Layer Security (WTLS), which provides authentication and confidentiality.

The WML markup language is based on XML. It can be viewed as a simplified version of HTML. It allows the user to enter data, select from a list, and display text and certain types of images. Example 10-1 shows a simple WML code sample.

Example 10-1 A simple WML code sample

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="card1" title="A first example">
    <p align="center">
      Hello WAP World.
    </p>
  </card>
</wml>
```

10.2 Deploying the WAP portal adapter

During the IBM Commerce Enhancement Pack - April 2003 Edition installation, the <WC_HOME>\schema\db2\wcs.portal.sql script is executed. This SQL script updates the WebSphere Commerce instance database tables PVCDEVMDL, PVCDEVSPEC, and PVCMDLSPEC for the portal and WAP portal adapter.

By default, wcs.portal.sql script set the deviceFormatId to the next available entry number. You must ensure the MODEL_ID in the PVCDEVMDL table updated by the wcs.portal.sql script matches the deviceFormatId in the WebSphere Commerce instance XML file (Example 10-2 on page 351).

To enable the WebSphere Commerce WAP portal adapter, do the following:

1. Ensure the wcs.portal.sql file has been executed. When installing the IBM Commerce Enhancement Pack - April 2003 Edition, this SQL file is executed by the EP1_DBUpdate.db2 script. If you have executed the EP1_DBUpdate.db2 script as documented in the installation procedure, this step has already been completed.
2. Determine the MODEL_ID in the PVCDEVMDL table for the adapter (PORTAL and WAP PORTAL). For example, to query the database for the PORTAL adapter, enter the following from a DB2 command window:

```
db2 connect to <wc_dbname>
db2 select * from pvcdevmdl
```

Record the value of the MODEL_ID for the desired adapter.

Update the WebSphere Commerce instance XML file found in the <WC_HOME>\instances\<instance>\xml\<instance>.xml directory with a text editor.

- a. Update deviceFormatId.
 - i. Search deviceFormatId within the HttpAdapter tag for the WAP portal the <instance>.xml file.
 - ii. Update the value of the deviceFormatId to the MODEL_ID recorded for the adapter.
- b. Enable the adapter.

Enable the deployment descriptor of the WebSphere Commerce portal adapter by changing the enabled="false" value to enabled="true" in the <instance>.xml file as seen in Example 10-2.

Example 10-2 Enable the WebSphere Commerce WAP portal adapter

```
<HttpAdapter deviceFormatId="-3"
  enabled="true"
  deviceFormatType="PVCAdapter"
  deviceFormatTypeId="-1"
  name="WAPORTAL">
  <factoryClassname="com.ibm.commerce.portal.adapters.WapPortalPvcAdapter">
  <PVCAdapter preferredLogonTimeout="20"
  bufferTimeout="5"
  registrationMode="0"/>
</HttpAdapter>
```

Enable the deployment descriptor of the WebSphere Commerce mobile portal adapter by changing the enabled="false" value to enabled="true" in the <instance>.xml file as seen in Example 10-2.

- c. Save and close file.

When complete with the changes to the <instance>.xml file, save the file and restart the WebSphere Commerce <instance> application server.

10.3 OpenWave.SDK 6.2 for WAP toolkit and simulator

When developing a mobile application, you will need a combination of toolkits, software simulators, and real mobile device hardware for testing. We specifically selected a OpenWave-based simulator to explore unique issues related to the microbrowsers.

We used the OpenWave.SDK 6.2 for WAP when developing the WAP sample code in this redbook.

OpenWave Phone.com, formally Unwired Planet, provides a Software Developers Kit (SDK) that includes a simulator that supports WAP content by its microbrowser, examples, and supporting documentation.

Although Nokia is dominant in Europe, the UP Phone.com browser is used by many phone manufacturers.

There are currently two versions of the OpenWave SDK that support WAP content:

- ▶ OpenWave SDK V6.1 for WAP
- ▶ OpenWave SDK V6.2 for WAP

Download and install OpenWave SDK for WAP

You can download the OpenWave SDK V6.2 for WAP from the following URL:

<http://developer.openwave.com/download/index.html#sdk>

Configure the OpenWave SDK mobile client

After you have installed and configured your mobile client, make sure that you define one of the following user agent ID:

- ▶ WAP
- ▶ Wap
- ▶ Nokia
- ▶ UP/4
- ▶ Rainbow

Note: At the time of writing this redbook, we found that the WAP portal adapter contained the static user agents listed and did not include a properties file or other means to add new user agents.

To work around this problem for development and test purposes, we modified the OpenWave simulator configuration file `DEVICEID` to `WAP`, which is one of the device types the WAP portal adapter does recognize.

To set the user agent ID do the following:

1. Open the mobile configuration file of your mobile client software development kit.
2. Modify the user agent ID and set the value to `WAP`, `Wap`, `Nokia`, `UP/4`, or `Rainbow`. For example:
`DEVICEID WAP`
3. Save your changes.
4. Restart the mobile client simulator.
5. Access your WebSphere Portal site. For example:
`http://<host_name>/wps/portal`

10.4 Sample WAP WML commerce portlet JSPs

The ITSO provided workspace includes WAP WML commerce portlet JSPs for the ITSO B2B CEP store. We created the following JSPs for the sample, which can be found in the workspace at `b2bCepWeb\webApplication\wml_portal_jsp`:

- ▶ `CategoryDisplay.jsp`
- ▶ `getResource.jsp`
- ▶ `subcategory.jsp`
- ▶ `topcategory.jsp`

We did not convert all the portlet JSPs from HTML to WML. The objective is to provide a basic sample.

Note: For information on WAP and WML, we recommend *Professional WAP*.



Appendixes



WebSphere Portal and WebSphere Commerce sample LDIF files

This appendix includes sample LDIF files for WebSphere Portal and WebSphere Commerce used to import data into the IBM Directory Server database.

The appendix includes the following sections:

- ▶ WebSphere Portal LDIF sample
- ▶ WebSphere Commerce LDIF sample

WebSphere Portal LDIF sample

Example A-1 contains a sample WebSphere Portal LDIF file used to import into the IBM Directory Server database for the ITSO working example.

Note: The LDIF sample file `wp-itso.ldif`, listed in Example A-1 can be found in the `c:\sg246890-code\ldif\wp-itso.ldif` directory.

This file is based on the WebSphere Portal LDIF template included with WebSphere Portal.

Example: A-1 WebSphere Portal LDIF sample (wp-itso.ldif)

version: 1

NOTE: you must edit this file before importing it and replace all
occurrences of the default suffix "dc=ibm,dc=com" with the suffix
that your LDAP server is configured for.

```
dn: dc=ibm,dc=com
objectclass: domain
objectclass: top
# Add lines according to this scheme that correspond to your suffix
dc: ibm,dc=com
dc: ibm
```

```
dn: cn=users,dc=ibm,dc=com
objectclass: container
objectclass: top
cn: users
```

```
dn: cn=groups,dc=ibm,dc=com
objectclass: top
objectclass: container
cn: groups
```

```
dn: uid=wsadmin,cn=users,dc=ibm,dc=com
objectclass: organizationalPerson
objectclass: person
objectclass: top
objectclass: inetOrgPerson
uid: wsadmin
userpassword: {iMAsK}>16LcsthmrncZ0VioPS+CLJI+MYQilJ7QNwPlmx8nPPfdak794823fyV
9dHh8wk76EcRs9NSrNN9PIK6pqNSs/gnaMbPvTkAp98J9uquduf8McUdpj6Bdjpcpe/A18d0e3TM
anELn9qKYdyzan/rE/ksPyMvQ1Dv9<
sn: admin
givenName: wps
```


cn: wps admin

```
dn: uid=wpsbind,cn=users,dc=ibm,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
uid: wpsbind
userpassword: {IMASK}>1As640BgqGS0YRnxNxa/YZbxY0H29yF9zM+Zq[4C53TG6RvCko5DnYEH0
8PC7jFc5i100nY1Fn54FE2Ft1c/1n3z4tUfNGYrk11iuwksTTelU/xZM00YfLQe+y7km8QsEMoZFp
qrtsysnpjYvYeVodYZSD6i15iKL6H4<
sn: bind
givenName: wps
cn: wps bind

dn: cn=wpsadmins,cn=groups,dc=ibm,dc=com
objectclass: groupOfUniqueNames
objectclass: top
uniquemember: uid=wpsadmin,cn=users,dc=ibm,dc=com
cn: wpsadmins
```

WebSphere Commerce LDIF sample

Example A-2 contains a sample WebSphere Commerce LDIF file used to import into the IBM Directory Server database for the ITSO working example.

Note: The LDIF sample file `wp-itso.ldif`, listed in Example A-2 can be found in the `c:\sg246890-code\ldif\wc-itso.ldif` directory.

This file is based on the WebSphere Commerce sample LDIF file included with the IBM Commerce Enhancement Pack - April 2003 Edition.

Example: A-2 WebSphere Commerce LDIF sample (wc-itso.ldif)

version: 1

```
dn: dc=ibm,dc=com
objectclass: domain
objectclass: top
objectclass: organization
dc: ibm,dc=com
dc: ibm
o: Root Organization
```

dn: cn=users,dc=ibm,dc=com
objectclass: container
objectclass: top
objectclass: organization
cn: users
o: Default Organization

dn: cn=groups,dc=ibm,dc=com
objectclass: top
objectclass: container
cn: groups

dn: uid=wcsadmin,dc=ibm,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: ePerson
objectclass: inetOrgPerson
givenname: WCS
uid: wcsadmin
userpassword:
sn: Admin
cn: Admin/WCS



Logon for commerce portlets without single sign-on

This appendix describes how to enable logon of commerce portlets without single sign-on within the commerce enabled portal development environment detailed in Chapter 6, "Implement the development environment" on page 253.

Architectural decisions

This section lists the ITSO architectural decisions for addressing the need for a debug development environment between WebSphere Portal and WebSphere Commerce by enabling logon of commerce portlets without single sign-on.

Table B-1 AD01: Modify commerce portlet base class

Architectural Decision ID	AD01
Architectural Decision	Modify commerce portlet base class to transparently call WebSphere Commerce logon command to create WebSphere Commerce session when single sign-on is not available.
Problem Statement	WebSphere Studio Application Developer test environment only runs on WebSphere Application Server Single Server Edition, which does not support LDAP or single sign-on. When running WebSphere Studio Application Developer test environment, LDAP and single sign-on are also not supported. An alternative authentication mechanism is necessary to enable debugging of commerce enabled portlets and JSPs.
Assumptions	WebSphere Studio Application Developer is used for commerce portlet development (WebSphere Portal), commerce development and debugging.
Motivation	Enable debugging for commerce enabled portlets and JSPs.
Alternatives	<ol style="list-style-type: none">1. Modify commerce portlet base class to transparently call WebSphere Commerce logon command to create WebSphere Commerce session when single sign-on is not available.2. Deploy portlets on WebSphere Application Server Advanced Edition with LDAP and single sign-on enabled. Use remote Java debugger.
Decision	Implement WebSphere Studio Application Developer 5 environment with both WebSphere Commerce and WebSphere Portal.

Table B-2 AD02: Use WebSphere Portal user ID for WebSphere Commerce logon

Architectural Decision ID	AD02
Architectural Decision	Use WebSphere Portal user ID for WebSphere Commerce logon.
Problem Statement	A user ID is needed to call logon command in WebSphere Commerce
Assumptions	Single sign-on is not used.
Motivation	Allow debugging of portlets with several users having different roles.
Alternatives	<ol style="list-style-type: none"> 1. Obtain user ID from portal User object and use it when issuing WebSphere Commerce logon command. 2. Read WebSphere Commerce user ID from configuration file. 3. Store WebSphere Commerce user IDs and passwords in WebSphere Portal credential vault.
Decision	The first alternative was selected due to greater flexibility than second alternative, and smaller amount of code modifications compared to third alternative.

Table B-3 AD03: Store WebSphere Commerce logon URL in a property file

Architectural Decision ID	AD03
Architectural Decision	Store WebSphere Commerce logon URL in a property file together with other WebSphere Portal property files.
Problem Statement	The WebSphere Commerce logon URL needs to be known to commerce portlets.
Assumptions	All commerce portlets are accessing the same WebSphere Commerce server, so a single logon URL can be shared across all portlets.
Motivation	
Alternatives	<ol style="list-style-type: none"> 1. Store WebSphere Commerce logon URL in a property file. 2. Provide the logon URL as portlet configuration parameter that can be read via PortletSettings.

Architectural Decision ID	AD03
Decision	The first alternative was chosen because the second scenario requires adding a configuration parameter for every commerce portlet.

Table B-4 AD04: Store WebSphere Commerce logon password in a property file

Architectural Decision ID	AD04
Architectural Decision	Store WebSphere Commerce logon password in a property file together with other WebSphere Portal property files.
Problem Statement	A password is needed to call logon command in WebSphere Commerce machine.
Assumptions	All users defined in WebSphere Commerce test system must have the same password.
Motivation	
Alternatives	<ol style="list-style-type: none"> 1. Store WebSphere Commerce password in property file. 2. Store WebSphere Commerce password in WebSphere Portal credential vault.
Decision	The first alternative was chosen to minimize development effort. To use a similar logon mechanism in production environment, WebSphere Portal credential vault must be used.

Modifications to Commerce Enhancement Pack code

The RemoteServletInvokerPortlet.java was modified to invoke the WCS Logon command if SSO is disabled. The modified code will use the current portal user ID to log on to WebSphere Commerce. Since in a development environment (WebSphere Application Server Advanced Single Server Edition) Java Authentication and Authorization Service (JAAS) functionality is limited, it is not possible to get the PasswordCredential object for the current user. To minimize code modifications, we decided that for development environment it is acceptable to have the same password for all WebSphere Commerce users. The password and WebSphere Commerce logon URL are stored in the WCSLogon.properties file located in C:\WebSphere\AppServer\lib\app\config\services\WCSLogon.properties.

When a new `HttpClient` object is created and the `SSOEnabled` flag is set to `false`, WebSphere Commerce logon command is called with the current portal user ID and the password read from `WCSLogon.properties` file. The HTTP response from the logon command contains WebSphere Commerce session cookies. The cookies are automatically stored in the `HttpClient` `CookieTable` object and included in subsequent requests to WebSphere Commerce. In this case, the WebSphere Commerce session is managed like a regular user browser session without the need for single sign-on capabilities.

Note: The `getHttpClientForThisApplicationSession()` method was static in the original code. It was modified to be nonstatic to be able to access `PortletContext` object.

Example: B-1 Modified RemoteServletInvokerPortlet.java code

```
private synchronized HttpClient
getHttpClientForThisApplicationSession(PortletRequest portletrequest) {
    com.ibm.commerce.portal.wpsapiextensions.ApplicationSession
applicationSession =
    ApplicationPortlet.getApplicationSession(portletrequest, true);
    Object obj = applicationSession.get("httpClientobjectkey");
    if (obj == null) {
        obj = new HttpClient();
        applicationSession.put("httpClientobjectkey", obj);

        if (!isSSOEnabled(portletrequest)) {
            try {
                PortletContext context = getPortletConfig().getContext();
                //Get the Logon URL and password from properties file
                String password =
                    context.getText(
                        "app.config.WCSLogon",
                        "WCSPassword",
                        portletrequest.getLocale());
                String logonURLTemplate =
                    context.getText(
                        "app.config.WCSLogon",
                        "WCSLogonURL",
                        portletrequest.getLocale());
                String userID = portletrequest.getUser().getUserID();

                StringBuffer logonURL = new StringBuffer(logonURLTemplate);
                logonURL.append("&logonId=").append(userID);
                logonURL.append("&logonPassword=").append(password);

                ((HttpClient) obj).getThisURLContents(
                    logonURL.toString(),
```

```

        portletrequest.getClient().getUserAgent(),
        new Hashtable());
    } catch (Exception e) {
        getPortletLog().error("Could not perform WCS Logon", e);
    }
}
}
return (HttpClient) obj;
}

private boolean isSSOEnabled(PortletRequest portletrequest) {
    // The SSOEnabled flag is not visible from this class, so we
    // get it from PortletSettings again
    PortletSettings portletSettings = portletrequest.getPortletSettings();
    String ssoString = portletSettings.getAttribute("SSOEnabled");
    boolean ssoboolean = false;

    if (ssoString != null) {
        ssoboolean = (new Boolean(ssoString)).booleanValue();
    }
    return ssoboolean;
}

```

Known limitations

The limitations of this solution have been identified as follows:

1. All WebSphere Commerce user IDs used for debugging should match WebSphere Portal user IDs.
2. All WebSphere Commerce user passwords should be the same and match the password defined in WCSLogon.properties file in the WebSphere Portal machine.
3. All portlets should have the SSOEnabled flag set to either true or false. Otherwise, results may be unpredictable.
4. All portlets must point to the same WebSphere Commerce machine.
5. The currently implemented code is designed for use in a development environment only, and has several security vulnerabilities. With some modifications it could be made secure for use in production environments where single sign-on between WebSphere Portal and WebSphere Commerce systems is not possible.

Updating the WebSphere Studio Application Developer plug-in with the modified classes

To update the WebSphere Studio Application Developer plug-in with the modified classes, do the following:

1. Export the Java project from WebSphere Studio Application Developer to WebsphereCommerceBasePortlet.jar file.
2. Copy the exported file to
<WSAD_HOME>\plugins\com.ibm.commerce.portal.tooling\data.
For example, in the ITSO development environment the full path of the final file location was as follows:
`c:\ibm\wsad\plugins\com.ibm.commerce.portal.tooling\data\WebsphereCommerceBasePortlet.jar`
3. Restart WebSphere Studio Application Developer.

Updating B2B Direct sample with the modified classes

To update the B2B Direct sample with the modified classes (or other samples), do the following:

1. Copy the exported WebsphereCommerceBasePortlet.jar to the commerceportal.jar file.
2. Go to C:\temp\CEP\B2BDirectPortal\CommerceRefApp\bin.
3. Extract WebsphereCommerceEnabledPortalB2BDirect.war if not already extracted:
`jar -xf WebsphereCommerceEnabledPortalB2BDirect.war`
4. Replace the WEB-INF/lib/commerceportal.jar file with the modified one.
5. If you will be using these portlets in a development environment, edit the portlet.xml file and change all SSOEnabled attributes to false (see Example B-2 on page 363).

Example: B-2 Update SSOEnabled attributes in portlet.xml

```
<config-param>
  <param-name>SSOEnabled</param-name>
  <param-value>false</param-value>
</config-param>
```

6. Recreate the WebsphereCommerceEnabledPortalB2BDirect.war file:

```
jar -cf WebsphereCommerceEnabledPortalB2BDirect.war WEB-INF
```

7. Deploy the portlets from the `WebSphereCommerceEnabledPortalB2BDirect.war` file.

Configuring WebSphere Commerce logon URL and password

To configure the WebSphere Commerce logon URL and password, do the following:

1. In the `<WAS_HOME>\lib\app\config` directory, create a file called `WCSLogon.properties` as shown in Example B-3. The `c:\sg246890-code\debug` directory contains a sample of this file.

Example: B-3 Sample WCSLogon.properties

```
WCSPassword=password
WCSLogonURL=http://<fully_qualified_hostname>:8080/webapp/wcs/stores/servlet/Logon?URL=http://<fully_qualified_hostname>:8080/webapp/wcs/stores/dummy.html&storeId=10051&reLogonURL=http://<fully_qualified_hostname>:8080/webapp/wcs/stores/dummy.html
```

2. Set `WCSPassword` property to your WebSphere Commerce password.
3. Set `WCSLogonURL` property to your WebSphere Commerce logon URL without the `logonId` and `logonPassword` parameters (change host name). `logonId` and `logonPassword` will be automatically appended to the URL.



Tips and troubleshooting for commerce enabled portals

This appendix includes troubleshooting techniques and tips for WebSphere Commerce and WebSphere Portal.

The appendix includes the following topics:

- ▶ Resolving issues during installation or testing
- ▶ Troubleshooting a store publishing failure
- ▶ Back up a DB2 database
- ▶ Restore a DB2 database
- ▶ Reset a disabled account

Resolving issues during installation or testing

This section includes tips for resolving issues encountered during the installation of the commerce enabled portal runtime and development environment, and during testing.

Cannot start portal using Application Developer

There are several reasons you may not be able start the portal when using WebSphere Studio Application Developer:

- ▶ Depending on the error you receive, there could be numerous reasons for this error to occur:
 - The IBM Agent Controller or WebSphere Studio Application Developer may have been installed prior to WebSphere Application Server
 - You may have multiple installations of WebSphere Application Server on the machine
 - Depending on when WebSphere Studio Application Developer or the IBM Agent Controller were installed, WebSphere Application Server may be configured to use the WAS_HOME environment variable, or it may be configured to point directly to the WebSphere installation directory.
 - If it is configured to use the WAS_HOME environment variable, then WebSphere Application Server may be having difficulties locating this variable or the variable might be incorrect.

A workaround is to modify the <AGENT_CONTROLLER_HOME>\config\serviceconfig.xml file. (This is typically located in <WSAD_HOME>\IBM Agent Controller. If WebSphere Application Server is in a different node from WebSphere Studio Application Developer, then this path will be where you have installed IBM Agent Controller.) Replace all entries of %WAS_HOME% with the location of your WebSphere Application Server installation (for example, C:\WebSphere\AppServer). Then, restart the IBM Agent Controller Service for the changes to take effect. When the IBM Agent Controller is restarted, restart the server.

Example: C-1 Application tag sample for serviceconfig.xml

```
<Application executable="java.exe"
  path="C:\websphere\wsad\jre\bin\java.exe" location="C:\\">
  <Variable name="CLASSPATH"
    value="C:\websphere\wsad\IBM Agent Controller\lib\picb.jar"
    position="append"/>
  <Variable name="CLASSPATH"
    value="C:\websphere\wsad\IBM Agent Controller\lib\logutil.jar"
```

```

        position="prepend"/>
</Application>
<Application executable="wteRemote.exe"
  path="%WAS_HOME%\java\bin\java.exe" location="%WAS_HOME%"
  <Variable name="CLASSPATH" value="
    C:\webSphere\wsad\IBM Agent Controller\lib\wasTools.jar;
    C:\webSphere\wsad\IBM Agent Controller\lib\wteServers.jar;
    C:\webSphere\wsad\IBM Agent Controller\lib\logutil.jar;
    %WAS_HOME%\properties;
    %WAS_HOME%\lib\bootstrap.jar"
    position="replace"/>
  <Variable name="PATH" value="%WAS_HOME%\bin" position="append"/>
</Application>

```

- ▶ Another problem could be that the IBM Agent Controller is not started on the WebSphere Application Server machine. If you receive an error fairly quickly stating that the server could not be started, then you should check to see if the IBM Agent Controller had started. The IBM Agent Controller is available as a Windows service on the machine that hosts WebSphere Application Server if it was installed properly.
- ▶ If you receive an error stating that port 900 or 9000 is already in use, then this might mean that WebSphere Application Server is started outside WebSphere Studio Application Developer. Make sure that the `java.exe` process is not running by checking the Task Manager in Windows or trying to stop the server using the `stopServer.bat` file in `<WAS_HOME>\bin`.
- ▶ If you receive numerous exceptions in the console and the server simply will not start, then possibly the Portal Server instance was not properly configured or you did not reboot the machine after the installation prior to starting this Portal instance. You should remove the Portal Server instance and configuration and recreate it following the steps provided previously, or reboot the machine and try again.

Portal cannot connect to the database during installation

Depending on the error, there could be multiple reasons for this to occur. Verify the following before attempting to retry the connection:

- ▶ Verify that the `<DB2_ROOT>\java12\usejdbc2.bat` file was executed properly so that Portal can locate the proper class files needed to connect to the database. Refer to the product installation help documentation.
- ▶ Verify that the user ID and password provided during the Portal installation is the same user ID and password used during the DB2 installation.
- ▶ Verify that the DB2 - DB2 service is started in the Windows services.

404 - File not found exception when accessing portal

This error would occur when the host name or port number specified during the configuration cannot be resolved or was entered incorrectly.

Also, if you are using the IBM HTTP Server, verify that it is started or restart it to verify that the WebSphere Application Server plug-in information is reloaded.

The system could not log into your account when running the portlet on the server

When running your portlet on the server, you might get an error indicating "The system could not retrieve your user account information data store. Please try again later." The WPSDEBUG ID is required to log into WebSphere Portal. Verify that the user ID and password are correct when you configured the Portal Server instance, and that the user ID was created correctly.

Error occurs when publishing the portlet application

Verify that the directory that you specified when configuring the Portal Server instance already exists. The Portal Toolkit will not create this directory for you.

Portlet page not found

This might occur when the deployment directory specified during the configuration does not match the location specified to copy the files. The Portal Toolkit would deploy the files, but would not be able to find the files when trying to display them. This might also occur if your debug ID or password does not match the one created earlier.

WebSphere Portal installation and configuration

When installing Portal on WebSphere Application Server Advanced Single Server Edition, it is important to note the following:

- ▶ To use the Portal Toolkit, you must install Portal on the Advanced Single Server Edition of WebSphere Application Server.
- ▶ An HTTP server is not required to use the Portal Toolkit.
- ▶ You must copy the `personalization.jar` and `prCommon.jar` files to the `<WAS_HOME>\lib\app` directory before installing Portal.
- ▶ You must apply WebSphere Application Server FixPak 2 and interim fix PQ56615 *before* installing Portal.

- ▶ Use the host name format of `http://yourserver.domain.com:9080` so that you do not need to regenerate the plug-in information while developing portlets.
- ▶ You must deploy the base portlets.
- ▶ Make sure to run `usejdbc2.bat`, found in the `<DB2_HOME>\java12` directory to enable the correct JDBC drivers for Portal.

Troubleshooting a store publishing failure

If you experience problems while publishing a store archive, review the troubleshooting techniques outlined in this section.

WebSphere Commerce log files

Review the following WebSphere Commerce log files found in the `<WC_HOME>\instances\<instance_name>\logs` directory:

- ▶ `message.txt`
- ▶ `trace.txt`
- ▶ `ecmsg_<hostname>_<timestamp>.log`
- ▶ `wcs.log`

Unlocking a store archive

A store archive may be locked during publishing because someone else is using it, or if an error occurs while accessing the store archive. Before unlocking the file, ensure that no one else is using the file.

To unlock a store archive, delete the lock file in the `<WC_HOME>\temp\demo\tools\devtools\lock` directory.

Republish a contract or account

If a message in the `ecmsg_<hostname>_<timestamp>.log` reports that publishing has failed while attempting to republish the contract or account asset, do the following before attempting to republish again:

- ▶ For contracts:
Ensure that you have changed the sequence number in the `contracts.xml` file to the next number in the sequence. The sequence number is a combination of the value for the `majorVersionNumber` and the `minorVersionNumber`.

For example if the `majorVersionNumber="1"`, and the `minorVersionNumber="0"`, change the value for the minor version number to 1. The resulting sequence number will be 1.1.

When you change the major version number and then republish the store archive, a new contract will be created in the database. This new contract replaces the previous contract.

If you do not change the major version number, a "Contract exists" error will be recorded in the logs when you republish the store archive, and the contract information will not be updated in the database

- For accounts:

Remove the `account.xml` file from the store archive and the `sarinfo.xml` file.

Publishing is successful but store does not display or does not function properly

If publish is stated to be successful, but you can't launch the store, or the store isn't functioning properly, check `ecmsg_hostname_timestamp.log` or `ecmsg_instancename_timestamp.log` (iSeries™) to ensure that the contract and account assets published properly. If not, see "Republish a contract or account" on page 373.

Transaction log for the database is full

If the message, "Transaction log for the database is full" displays in the `message.txt` log, you have the following options:

- Create secondary transaction logs using the DB2 Control Center. For more information, see the *DB2 Administration Guide*.
 - Increase the transaction log file size by doing the following:
 - a. In a DB2 command window, type:

```
db2 get db cfg for <wc_database>
```
 - b. Then, look for the log file size (`logfilsiz`). After finding the log file size, type:

```
db2 update db cfg for <wc_database> using <logfilsiz>
```

Where `<logfilsiz>` is a larger number than the previous number.
 - c. Restart DB2.
- or
- In a DB2 command window, type:
- ```
db2 update db cfg for <wc_database> using LOGPRIMARY 20
```



Where 20 is the number of primary logs (this number may be different for your site). Increasing the LOGPRIMARY increases your space requirement. For more information, see the *DB2 Administration Guide*.

- d. If you are using DB2, you can also drop the following summary tables, using the DB2 command **DB2 drop table xxx:**
- PRODUCT
  - STOREINV
  - CATEGORY
  - RICHATTR
  - RICHATTRCATGP

**Note:** The PRODUCT table is most likely to cause the Transaction log to overflow, so delete it first and try to publish the store before deleting the other tables.

Create secondary transaction logs and attempt to publish the store before deleting summary tables.

## java.lang.OutOfMemory error

If the publishing status in the Store archive list page remains at Publishing, and there is no computer activity indicating that the publish is still in progress, check the wcs.log file. If the message, "java.lang.OutOfMemoryError" or a similar message displays in the wcs.log file, increase the servlet JVM heap size in the WebSphere Application Server.

## Back up a DB2 database

To back up a DB2 database, do the following:

1. Stop the applications that are connected to the DB2 database.
2. Disconnect all applications connected to the database:  

```
db2 force applications all
db2 terminate
```
3. Create a database backup directory (for example, c:\ibm\dbbackup).
4. Open a DB2 command window.
5. Back up the database by entering the command:

```
db2 backup db <dbname> to <path>
```

For example:

```
db2 backup db wcldb to c:\ibm\dbbackup
```

## Restore a DB2 database

If you need to restore a DB2 database, do the following:

1. Stop the applications that are connected to the DB2 database.
2. If a database exists on the DB2 Server that is no longer needed, drop the database with the following command from a DB2 command window:
3. Copy the backup of the database to the DB2 Server.
4. To restore the database, enter the following from a DB2 command window:

```
db2 drop db <db_name>
```

```
db2 restore db <wc_dbname> from <path>
```

For example:

```
db2 restore db wc1db from c:\ibm\dbbackup
```

## Reset a disable account

In the event that you incorrectly enter your password and the account becomes disabled, do the following:

1. Open a DB2 command window.
2. Connect to the WebSphere Commerce instance database:
3. Enter the following DB2 command to reset the WebSphere Commerce wcsadmin login ID:

```
db2 connect to <db_name>
```

```
db2 update USERREG SET STATUS=1, PASSWORDRETRIES=0 WHERE LOGINID='wcsadmin'
```



## Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

### Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

`ftp://www.redbooks.ibm.com/redbooks/SG246890`

Alternatively, you can go to the IBM Redbooks Web site at:

`ibm.com/redbooks`

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG246890.

### Using the Web material

The additional Web material that accompanies this redbook includes the following files:

| <i>File name</i>    | <i>Description</i>                            |
|---------------------|-----------------------------------------------|
| <b>SG246890.zip</b> | WebSphere Commerce Portal zipped code samples |

## System requirements for downloading the Web material

The following system configuration is recommended:

|                          |                    |
|--------------------------|--------------------|
| <b>Hard disk space:</b>  | 40 MB minimum      |
| <b>Operating System:</b> | Windows NT or 2000 |
| <b>Processor:</b>        | 1 GHz or higher    |
| <b>Memory:</b>           | 768 MB or higher   |

## How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

## IBM Redbooks

For information on ordering these publications, see "How to get IBM Redbooks" on page 381.

- ▶ *WebSphere Commerce V5.4 Handbook*, SG24-6567
- ▶ *WebSphere Commerce V5.4 Developer's Handbook*, SG24-6190
- ▶ *WebSphere Portal V4.1, Windows 2000 Installation*, REDP3593
- ▶ *WebSphere Portal V4.1 Developer's Handbook*, SG24-6897
- ▶ *WebSphere Portal V4.1 Handbook Volume 1*, SG24-6883
- ▶ *WebSphere Portal V4.1 Handbook Volume 2*, SG24-6920
- ▶ *WebSphere Portal V4.1 Handbook Volume 3*, SG24-6921
- ▶ *Access Integration Pattern using IBM WebSphere Portal Server*, SG24-6267
- ▶ *Applying Pattern Approaches: Patterns for e-business Series*, SG24-6805
- ▶ *B2B e-commerce With WebSphere Commerce Business Edition V5.4, Patterns for e-business Series*, SG24-6194
- ▶ *A Portal Composite Pattern Using WebSphere Portal V4.1.2*, SG24-6869
- ▶ *Mobile Applications with IBM WebSphere Everyplace Access Design and Development*, SG24-6259
- ▶ *Integrating WebSphere Commerce V5.4 and WebSphere Portal V4.1.4*, REDP3684

## Other resources

These publications are also relevant as further information sources:

- ▶ *Portlet Developer's Guide, WebSphere Portal V4.1* found at:  
`ftp://ftp.software.ibm.com/software/webserver/portal/V41PortletDevelopmentGuide.pdf`

- ▶ IBM WebSphere Commerce product guides found at:  
[http://www.ibm.com/software/webservers/commerce/wc\\_be/11t-tech-general.html](http://www.ibm.com/software/webservers/commerce/wc_be/11t-tech-general.html)
  - *Fundamentals Guide, IBM WebSphere Commerce V5.4*
  - *Programmer's Guide, IBM WebSphere Commerce V5.4*
  - *Store Developer's Guide, IBM WebSphere Commerce V5.4*
  - *Installation Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Windows*
  - *Additional Software Guide, IBM WebSphere Commerce V5.4 Professional and Business Edition for Windows*
  - *Installation Guide, IBM WebSphere Commerce FixPak V5.4.0.5*
  - *Installation Guide, IBM WebSphere Commerce Studio V5.4 for Windows NT and Windows 2000*
- ▶ Jonathan Adams, et al., *Patterns for e-business: A Strategy for Reuse*, IBM Press, 2001, ISBN 1931182027
- ▶ Arehart, Charles, et al, *Professional WAP*, Wrox, July 2000, ISBN 1861004044
- ▶ *Pursuing efficiency and revenue with commerce enabled portals* found at:  
<http://www.ibm.com/software/webservers/commerce/portal/>

## Referenced Web sites

These Web sites are also relevant as further information sources:

- ▶ Patterns for e-business Web site:  
<http://www.ibm.com/developerWorks/patterns/>
- ▶ WebSphere Commerce V5.4, Business Edition  
[http://www.ibm.com/software/webservers/commerce/wc\\_be/11t-tech-general.html](http://www.ibm.com/software/webservers/commerce/wc_be/11t-tech-general.html)
- ▶ WebSphere Commerce V5.4, Professional Edition  
[http://www.ibm.com/software/webservers/commerce/wc\\_pe/11t-tech-general.html](http://www.ibm.com/software/webservers/commerce/wc_pe/11t-tech-general.html)
- ▶ OpenWave SDK V6.2 for WAP:  
<http://developer.openwave.com/download/index.html#sdk>
- ▶ WebSphere Portal InfoCenter:  
<http://publib.boulder.ibm.com/pvc/wp/42/smb/en/InfoCenter/index.html>

## How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

## IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.





# Index

## A

- Access control 57
  - WebSphere Commerce 57
  - WebSphere Commerce Portal 58
  - WebSphere Portal 57
- Access Integration business pattern 75
- Access Integration pattern 68–69
- Administration use cases 123
- Aggregation 7
- Aggregation Module 48
- Alert messaging portlet 36
- Ant 265
- ant\_create\_sar.xml 297
- APAR JR18068 198
- Application Integration pattern 68, 72
- Application pattern 75
- Architecture
  - WebSphere Commerce 37
  - WebSphere Commerce Portal 18
  - WebSphere Portal 46
  - working example 125
    - component model 132
    - conceptual model 135
    - decisions 128
    - operational model 134
    - overview 125
    - physical model 157
    - specified model 144
- Authentication 56
- Authentication Server 48

## B

- Backup DB2 database 375
- Business drivers 87
- Business scenario 100
  - challenges 101
  - initial context 100
  - requirements 102

## C

- CategoryDisplay.jsp 353
- cHTML 40

- Collaboration 50
- Collaboration business pattern 70
- Collaboration pattern 68
- Commerce enabled portlet
  - create store 285
  - debug logon without SSO 361
  - installation
    - WebSphere Application Server AEs 275
  - integration approaches 18
  - mobile client access 347
  - troubleshooting tips 369
- Commerce enabled portlet
  - WAP WML JSPs 353
- Commerce Enhancement Pack
  - installation 188
  - post-install configuration 191
- Component model 132
- Conceptual model 135
- Configuration
  - DB2 Client 249
    - Windows 2000 249
  - single sign-on 228
  - WebSphere Commerce for LDAP 210
  - WebSphere Portal 231
- content management 49
  - reference information 41
- Cookies 20
- crawl 90
- Create new organization 312
- Create SAR with Ant 297
- credentials 46

## D

- Database backup 196
- DB2 commands
  - attach 249
  - catalog TCP/IP node 249
  - list databases 178
- DB2 database backup 375
- DB2 restore database 376
- DB2 Server
  - implementation 245
- Deploy

- commerce enabled portal 309
  - development environment 332
  - runtime environment 310
- Developing commerce enabled portlets 291
- Development environment 253
  - architecture decisions for debug 362
  - configurations 258
    - multi-node 259
    - single-node 258
  - test runtime for team 260
  - hardware and software 261
- Device control 52
- deviceFormatId 350
- Directory Server node
  - implementation 199
    - configure WebSphere Commerce for LDAP 210
    - DB2 Server 200
    - enable SSL between WC and LDAP 214
    - IBM Directory Server configuration 202
    - IBM Directory Server installation 200
    - IBM HTTP Server 200
    - import LDIF 205
    - Windows 2000 Server 199
- Distinguished Name 206
  - uid=wcsadmin,dc=ibm,dc=com 360
- documents 89
- Domino Extended Search 90
- Dynamic Context Groups 29, 302
  - attributes in portlet.xml 302
  - create 301
  - defining 307
  - JSP modification 305
  - tooling 306
- Dynamic Context Portlet Group Assembly Tool 32

**E**

- ecmsg\_log 373
- Electronic Commerce composite pattern 67, 77
- Enable Portal URI 291
- Enable SSL
  - WebSphere Commerce and LDAP 214
- End-to-end development 254
  - solution overview 255
- enterprise application integration 46
- ePerson 242
- events 46
- Extended Enterprise business pattern 71

- Extended Enterprise pattern 68

## F

- Federated Search 90
- flat text 90
- Form naming conventions 291
- free-text 90
- Front-end experience use cases 116
- Functional requirements 102

## G

- getResource.jsp 353

## H

- HTML 349
- HttpAdapter 351
- HttpClient CookieTable object 365
- HttpServletRequest 19
- HttpServletResponse 19

## I

- IBM Commerce Enhancement Pack - April 2003 Edition 3
- IBM Directory Server installation 200
- i-mode 40
- Implementation
  - DB2 Server 245
  - IBM Directory Server 199
  - WebSphere Commerce node 173
  - WebSphere Portal 214
- Import LDIF data
  - WebSphere Commerce 207
  - WebSphere Portal 205-206
- Information Aggregation business pattern 71
- Information Aggregation pattern 68
- Installation
  - development environment
    - Commerce Enhancement Pack Toolkit 266
    - WebSphere Portal Server 4.2.1 268
    - WebSphere Portal Toolkit 4.2.5 268
    - WebSphere Studio Application Developer 5 262
  - IT drivers 87
  - ITSO B2B CEP store 100
    - create store archive 320
    - customization 291
    - prepare SAR for publish 319

- ITSO sample code 286
  - description 286
  - download procedure 286
  - zip file (sg246890.zip) 286
- ITSO scenario
  - publish store 321

## J

- Java Authentication and Authorization Service (JAAS) 364
- java.lang.OutOfMemory 375
- JavaScript naming convention 291
- JDBC2 175, 246

## L

- Layout 7
- LDAP 56, 210
  - create a new suffix 205
- ldapentry.xml 212, 242
- LDIF
  - WebSphere Commerce 359
  - WebSphere Portal 358
- Line-of-business 65
- LOB
  - See Line-of-business
- look and feel 46
- LtpaToken 20, 228–229, 231

## M

- manipulating 87
- Master portlet 303
- m-commerce
  - device control 52
  - security enhancements 53
  - session control 52
  - URL buffering 53
- Member subsystem 43
- message.txt 373
- Mobile client access 51
  - commerce enabled portal 347
- Mobile device challenges 52
  - device control 52
  - security 53
  - session control 52
  - URL length 53
- mobile devices 46

- Model-View-Controller (MVC) 21
- multiple devices 46
- MyPortalAlert 36

## N

- Non-functional requirements 105
  - availability 109
  - performance 106, 109
    - response time 106
  - static volumetric 107
  - throughput 106
  - utilization 107
  - system constraints 110
  - volumetric 111

## O

- OpenWave Phone.com 352
- OpenWave.SDK 6.2 for WAP 352
  - configuration 352
- Operational Data Store 90
- Operational model 134

## P

- Package a SAR with Ant 296
- Page 7
- Page rendering 61
- Patterns for e-business 64
  - introduction 64
  - layered asset model 66
  - Product mapping 94
  - Runtime patterns 92
  - Web site 67
- Permissions 7
- personalization 7, 46, 49
- Physical model 157
- Place 7
- Planning for implementation
  - development environment 253
  - runtime environment 166
- population 88
- Population Crawl and Discovery application pattern 88
- Population-Single Step application pattern 86
- Portal
  - benefits
    - customer loyalty 8
    - improve decision making 8

- marketing opportunities 9
    - productivity and efficiency 8
    - reach and user experience 9
  - defined 4
  - page 7
  - page group 7
  - page layout 7
  - permissions 7
  - place 7
  - services 4
  - skins 7
  - themes 7
  - Portal Engine 48
  - portal implementation 90
  - Portal Servlet 48
  - Portal/Commerce characteristics 72
  - portals
    - database structures 49
    - infrastructure 49
    - Jetspeed implementation 47
    - LDAP 49
    - Page content 47
    - page structure 47
    - Portal Engine 48
    - Portal Framework 46
    - portlets 48
    - profile information 46
    - services 49
    - Tooling 50
    - Web-based content 46
  - Portlet 5
  - Portlet application 6
  - Portlet page not found error 372
  - portlet.xml 302, 325
  - PortletResponse 19
  - portlets 46
  - Presentation services 47, 61
  - Product mapping
    - commerce enabled portal 95
  - Product mappings 94
  - PvC adapter 38
  - PvC commands 40
    - PvC data beans 40
    - PvC database tables 41
  - PvC data beans 40
  - PvCBufferDataBean 40
  - UserPvCDeviceDataBean 41
  - PvC database tables 41
    - reference information 41
  - PVCBINDING 41
  - PVCBUFFER 41
  - PvCBufferDataBean 40
  - PvCBufferUrl 40
  - PvCChangeDevice 40
  - PVCDEVMDL 41, 350
  - PVCDEVSPEC 41, 350
  - PVCMDLSPEC 41, 350
  - PvCRegistration 40
  - PvCRegistrationDevice 40
  - PVCSESSION 41
- ## R
- Redbooks Web site 381
    - Contact us xvi
  - ReEnterPassword 40
  - remote content 46
  - RemoteServletInvokerPortlet.java 364–365
  - Republish a contract or account 373
  - RequestDispatcher 19
  - Requirements 102
    - functional 102
    - non-functional 105
  - Reset disabled account 376
  - Resource Engine 7
  - Restore DB2 database 376
  - Role-based dynamic aggregation 35
  - Rule-based dynamic aggregation 33
  - Rules Engine 7
  - Runtime environment
    - implementation 165
    - WebSphere Commerce 37
    - WebSphere Commerce Portal 22
  - Runtime patterns 92
    - Commerce enabled portal
    - runtime pattern 94
- ## S
- Sametime integration 36
  - Search 50, 90
  - Search & Indexing node 90
  - security 46
  - Security enhancement 53
  - Self-Service business pattern 70
  - Self-Service pattern 68
  - ServletInvokerPortlet 18
  - Session control 52
  - Session management 59

- WebSphere Application Server 59
- WebSphere Commerce 59
- WebSphere Commerce Portal 60
- WebSphere Portal 60
- Short Message Services
  - See SMS
- Single sign-on 56, 228, 231, 237
  - scenario 229
  - WebSphere security 231
- Single sign-on (SSO) 8, 228
- Single Sign-On Application pattern 75
- Site Analysis 50
- Skins 7
- Slave portlets 304
- SMS 36
- Specified model 144
- store.xml 288
- Stripping HTML tags 291
- subcategory.jsp 353
- System use cases 122
- systems 87

## T

- Themes 7
- Tivoli Access Manager (TAM) 57
- Toolkit 51
- topcategory.jsp 353
- trace.txt 373
- Transaction log full 374
- Transcoding Technology 49
- transformation 86, 88
- Transient cookie 228
- Troubleshoot publishing
  - java.lang.OutOfMemory 375
  - log files 373
  - republish contract or account 373
  - transaction log full 374
  - unlock store archive 373
- Troubleshooting techniques 369
- Troubleshooting tips
  - cannot connect to db 371
  - cannot logon 372
  - cannot start portal 370
  - DB2 database backup 375
  - file not found exception 372
  - portlet not displayed 372
  - reset disabled account 376
  - resolving install issues 370

- restore DB2 database 376

## U

- Unlocking a store archive 373
- Unwired Planet 352
- URL buffering 53
- Use case model 113
- Use case summary 114
- Use cases
  - administration 123
  - front-end experience 116
  - system 122
- UserPVCDeviceDataBean 41

## W

- WAP 40, 347, 349
- WAP portal adapter 350
  - deviceFortalId 350
  - PVCDEVMDL 350
  - PVCDEVSPEC 350
  - PVCMDLSPEC 350
- WAP WML commerce portlet JSPs 353
- wc-itsc.ldif 359
- WCS V5.1
  - device control 52
  - session control 52
- wcs.log 373
- wcs.portal.sql 350
- WCSLogon.properties 364
- WCSLogonURL 368
- WCSPassword 368
- Web services 50
- WebSphere Application Server 177
  - installation 177
  - session management 59
- WebSphere Commerce
  - access control 57
  - architecture 37
  - configuration for LDAP 210
  - disable cache 212
  - enable portal adapter 196
  - HTTP request flow 44
  - implementation 173
    - APAR JR18068 198
    - Commerce Enhancement Pack 188
    - Commerce Enhancement Pack post-install config 191
    - database backup 196

- DB2 Server 174
- disable WebSphere Commerce cache 197
- enable portal adapter 196
- IBM HTTP Server 176
- instance creation 190
- tools verification 193
- WebSphere Application Server 177
- WebSphere Commerce 181
- WebSphere Commerce Payments 185
- Windows 2000 Server 174
- installation 181
- instance creation 190
- ldapentry.xml 212
- LDIF file 359
- log files
  - ecmsg\_.log 373
  - messages.txt 373
  - trace.txt 373
  - wcs.log 373
- Ligon URL controller command 257
- m-commerce direct
  - HTTP request flow 39
- m-commerce features
  - PvC adapter framework 40
  - PvC commands 40
- mobile client access 38
- runtime environment 37
- session management 59
- subsystems
  - Member subsystem 43
  - subsystems and components 41
  - tools verification 193
- WebSphere Commerce logon URL 368
- WebSphere Commerce Payments
  - configuration 194
  - installation 185
- WebSphere Commerce Payments FixPak V3.1.3 187
- WebSphere Commerce Payments FixPak V3.1.3.1 187
- WebSphere Commerce Portal
  - access control 58
  - architecture 18
  - component architecture 23
  - features 9
    - Dynamic Context Groups 29
    - real-time communication and collaboration 35
    - role-based dynamic aggregation 35
    - rule-based dynamic aggregation 33
  - HTTP request flow 26
  - integration architecture 25
  - mobile architecture 54
    - HTTP request flow 54
  - MyStore home page 6
  - packaging 11
  - prerequisite knowledge 12
  - runtime environment 22
  - session management 60
- WebSphere Commerce Server components 42
- WebSphere Personalization 7, 49
  - Resource Engine 7
  - Rules Engine 7
- WebSphere Portal 48–49
  - aggregation 7
  - aggregation modules 48
  - architecture 46
  - authentication server 48
  - configuration 231
  - Editions
    - Enable 4
    - Experience 4
    - Extend 4
  - home page 5
  - implementation 214
    - Commerce Enhancement Pack configuration 225
    - DB2 Server 215
    - IBM HTTP Server 216
    - WebSphere Application Server 216
    - WebSphere Portal installation 217
    - Windows 2000 Server 215
  - infrastructure 49
  - installation 217
  - key concepts 4
    - collaboration 4
    - enterprise application integration 4
    - mobile device support 4
    - personalization 4
    - security 4
    - single sign-on 4
    - Web content publishing 4
  - LDIF file 358
  - personalization 7
  - Portal Engine 48
  - portal servlet 48
  - portlet container 48
  - presentation services 47

- session management 60
- single sign-on 231, 237
- WebSphere Portal Content Publisher
  - personalization 7
- WebSphere Portal Server V4.2.1 263
- WebSphere Portal Toolkit V4.2.5 263
- WebSphere Studio Application Developer 261
  - configuration
    - edit large XML files 264
  - installation 262
  - PTF 001 261
- WebsphereCommerceBasePortlet.jar 367
- Wireless Application Protocol
  - See WAP
- Wireless Markup Language
  - See WML
- WML 40, 349
- WML code sample 350
- WMLScript 349
- Working example
  - architecture 125
  - business scenario 100
  - requirements 102
  - runtime implementation
    - hardware and software 167
    - high level procedure 172
    - network environment 166
  - use case model 113
- wp-itso.kdif 358

## X

- XML-based commerce portal 21







Redbooks

## WebSphere Commerce Portal V5.4 Solutions

(0.5" spine)  
0.475" <-> 0.875"  
250 <-> 459 pages







**Redbooks**

# WebSphere Commerce Portal V5.4 Solutions

**Integrating WebSphere Commerce V5.4, Business Edition and WebSphere Portal V4.2**

**WebSphere Commerce portal architecture and design guidelines**

Portals provide the user with a single point of access to a wide variety of content, data and services throughout an enterprise. The content displayed in portlets on the portal page can be personalized based on user preferences, site design and marketing campaigns.

**Implement a commerce portal runtime and development environment**

IBM WebSphere Commerce Portal provides an integrated solution for aggregating and personalizing commerce content and data by leveraging the technologies of WebSphere Commerce V5.4 and WebSphere Portal V4.2. When using the WebSphere Commerce Portal, users have a single point of access to personalized content in portlets from Web browser clients and mobile devices.

**Working example for developing a commerce enabled portal B2B site**

This IBM Redbook provides an overview of the WebSphere Commerce Portal benefits, features and architecture. The focus of the redbook is an end-to-end working example scenario for a commerce enabled portal B2B store. The working example includes the following topics: business requirements analysis and solution design, multi-tiered runtime implementation procedures, development environment with source level debug of commerce portlets and JSPs, create a commerce enabled portal store using WebSphere Studio Application Developer, create Dynamic Context Groups, and mobile client access.

**INTERNATIONAL  
TECHNICAL  
SUPPORT  
ORGANIZATION**

**BUILDING TECHNICAL  
INFORMATION BASED ON  
PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:  
[ibm.com/redbooks](http://ibm.com/redbooks)**